



TUGAS AKHIR – SS141501

KOMPARASI METODE EM-GMM (*EXPECTATION MAXIMIZATION-GAUSSIAN MIXTURE MODEL*) DAN FCM (*FUZZY C-MEANS*) DALAM SEGMENTASI CITRA OTAK MRI (*MAGNETIC RESONANCE IMAGING*) DI RSUD SOETOMO DALAM MENENTUKAN LOKASI TUMOR OTAK

WIN HEBER GOKLAS SIANIPAR
NRP 1313100 134

Dosen Pembimbing
Prof. Drs. Nur Iriawan, M.Ikom, Ph. D
Dr. Kartika Fithriasari, M.Si

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM



TUGAS AKHIR – SS 141501

KOMPARASI METODE EM-GMM (*EXPECTATION MAXIMIZATION-GAUSSIAN MIXTURE MODEL*) DAN FCM (*FUZZY C-MEANS*) DALAM SEGMENTASI CITRA OTAK MRI (*MAGNETIC RESONANCE IMAGING*) DI RSUD SOETOMO DALAM MENENTUKAN AREA TUMOR OTAK

**WIN HEBER GOKLAS SIANIPAR
NRP 1313 100 134**

**Dosen Pembimbing
Prof. Drs. Nur Iriawan, M.Ikom, Ph. D
Dr. Kartika Fithriasari, M.Si**

**PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN
ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017**



FINAL PROJECT – SS 141501

**COMPARISON BEETWEN EM-GMM (EXPECTATION
MAXIMIZATION-GAUSSIAN MIXTURE MODEL) FOR
BRAIN MRI SEGMENTATION IN RSUD SOETOMO TO
LOCATED BRAIN TUMOUR AREA**

**WIN HEBER GOKLAS SIANIPAR
NRP 1313 100 134**

**Supervisor
Prof. Drs. Nur Iriawan, M.Ikom, Ph. D
Dr. Kartika Fithriasari, M.Si**

**UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017**

LEMBAR PENGESAHAN

KOMPARASI METODE EM-GMM (*EXPECTATION MAXIMIZATION-GAUSSIAN MIXTURE MODEL*) DAN FCM (*FUZZY C-MEANS*) DALAM SEGMENTASI CITRA OTAK MRI (*MAGNETIC RESONANCE IMAGING*) DI RSUD SOETOMO DALAM MENENTUKAN AREA TUMOR OTAK TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Sains
pada

Program Studi Sarjana Departemen Statistika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember


Oleh:
Win Heber Goklas Sianipar
NRP. 1313 100 134

Disetujui oleh Pembimbing Tugas Akhir:

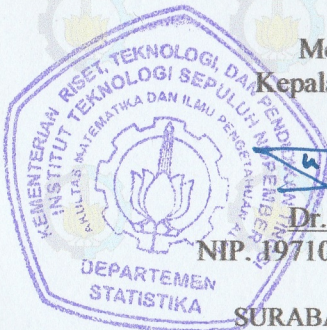
Prof. Drs. Nur Iriawan, M.Ikom, Ph. D
NIP. 19621015 198803 1 002

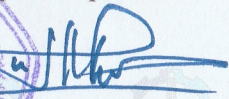
()

Dr. Dra. Kartika Fitriyasari, M. Si
NIP. 19691212 199303 2 002

()

Mengetahui,
Kepala Departemen




Dr. Suhartono
NIP. 19710929 199512 1 001

SURABAYA, JULI 2017

**KOMPARASI METODE EM-GMM (*EXPECTATION
MAXIMIZATION-GAUSSIAN MIXTURE MODEL*) DAN FCM
(*FUZZY C-MEANS*) DALAM SEGMENTASI CITRA OTAK MRI
(*MAGNETIC RESONANCE IMAGING*) DI RSUD SOETOMO
DALAM MENENTUKAN AREA TUMOR OTAK**

Nama Mahasiswa	: Win Heber Goklas Sianipar
NRP	: 1313 100 134
Departemen	: Statistika FMIPA - ITS
Dosen Pembimbing	: Prof. Drs. Nur Iriawan, M.Ikom, Ph. D Dr. Kartika Fithriasari, M.Si

Abstrak

Segmentasi citra sudah populer dilakukan, terutama untuk para peneliti di bidang Biomedis maupun Teknik Informatika. Segmentasi citra dalam bidang kesehatan, mempunyai tujuan utama untuk menentukan atau mendeteksi dini area tumor, salah satu citra medis yang populer saat ini adalah MRI. Penelitian ini, akan dilakukan studi data citra MRI di Rumah Sakit Umum Daerah Soetomo, Surabaya. Metode segmentasi yang dipakai adalah klastering menggunakan FCM dan EM-GMM. Dimana inisialisasi jumlah klaster ditentukan berdasarkan Silhouette Index untuk EM-GMM dan Partition Coefficient Index Untuk FCM. Selain berdasarkan nilai index tersebut penentuan dilakukan berdasarkan pengamatan subjektif dari pihak medis. Dalam komparasinya berdasarkan nilai similaritynya didapatkan metode EM-GMM lebih robust terhadap Salt and Pepper Noise dibanding FCM dan FCM lebih robust terhadap Gaussian Noise dibanding EM-GMM.

Kata kunci: *EM, FCM, GMM, Klastering, MRI, Optimasi, Partition Coefficient Index, RSUD Soetomo, Segmentasi, Silhouette Index, Similarity, Statistika Fitur Citra, Tumor*

**COMPARISON BEETWEN EM-GMM
(EXPECTATIONMAXIMIZATION-GAUSSIAN MIXTURE
MODEL) FOR BRAIN MRI SEGMENTATION IN RSUD
SOETOMO TO LOCATED BRAIN TUMOUR AREA**

Name : Win Heber Goklas Sianipar
NRP : 1313 100 134
Department : Statistics FMIPA - ITS
Supervisor : Prof. Drs. Nur Iriawan, M.Ikom, Ph. D
Dr. Kartika Fithriasari, M.Si

Abstract

Image segmentation has been popular, especially for researchers in the field of Biomedical and Informatics Engineering. Image segmentation in the health field, has the primary goal of determining or detecting early tumor areas, one of the most popular medical images currently is MRI. This research, will be studying MRI image data at RSUD Soetomo, Surabaya. The segmentation method used is clustering using FCM and EM-GMM. Where initialisation of cluster number is determined based on Silhouette Index for EM-GMM and Partition Coefficient Index For FCM. In addition to the index value based on the determination is done based on subjective observations from the medical party. In comparation based on the value of similaritynya obtained a more robust EM-GMM method against Salt and Pepper Noise than FCM and FCM more robust against Gaussian Noise than EM-GMM.

Keywords: *EM, FCM, GMM, Clustering MRI, Partition Coefficient Index, RSUD Soetomo, Segmentasi, Silhouette Index, Similarity, Image Statistic Features, Tumour*

KATA PENGANTAR

Puji syukur atas berkat Tuhan yang melampaui segala akal pikiran, yang dari pada-Nya diberikan damai sejahtera dan sukacita sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul **“Komparasi Metode EM-GMM (*Expectation Maximization-Gaussian Mixture Model*) dan FCM (*Fuzzy C-Means*) dalam Segmentasi Citra Otak MRI (*Magnetic Resonance Imaging*) di RSUD Soetomo dalam menentukan Area Tumor Otak”** dengan baik adanya.

Keberhasilan penyusunan Tugas Akhir ini tidak lepas dari banyaknya bantuan dan dukungan yang diberikan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Dr. Suhartono selaku Ketua Departemen Statistika dan Bapak Dr. Sutikno, M.Si selaku Koordinator Program Studi S1 yang telah memberikan fasilitas untuk kelancaran penyelesaian Tugas Akhir.
2. Bapak Prof. Drs, Nur Iriawan, M. Ikom, Ph. D selaku dosen pembimbing, dan Ibu Dr. Kartika Fithriasari, M.Si selaku dosen pembimbing sekaligus dosen wali yang telah dengan sabar memberikan bimbingan, saran, dan dukungan selama penyusunan Tugas Akhir.
3. Ibu Dr. Irhamah, S.Si., M. S.Si dan Ibu Dr. Santi Wulan, S. Si, M.Si selaku dosen penguji yang telah memberikan banyak bantuan dan saran untuk kesempurnaan Tugas Akhir ini.
4. Ibu Widiyana Ferriastuti, dr., Sp. Rad sebagai pembimbing medis penulis dalam penelitian ini yang tidak ada henti-hentinya memberikan waktunya untuk berdiskusi sehingga terselesaikannya Tugas Akhir ini.
5. Seluruh dosen Statistika ITS yang telah memberikan ilmu dan pengetahuan yang tak ternilai harganya, serta segenap karyawan Departemen Statistika ITS.

6. Bapak Sianipar dan Ibu Dr. Gultom selaku orang tua penulis, Yoko dan Yohanna yang merupakan Abang dan Adik penulis yang selalu memberikan dukungan, kasih sayang dan doa yang tidak pernah putus, serta senantiasa menjadi penyemangat bagi penulis dalam menyelesaikan Tugas Akhir ini.
7. Bu Anindya sebagai rekan diskusi penulis dalam penelitian ini yang tidak ada henti-hentinya memberikan waktunya untuk berdiskusi sehingga dapat diselesaikannya Tugas Akhir ini.
8. Rekan-rekan Mahasiswa Bona Pasogit (MBP) serta bang Saut Gurning selaku pembina MBP yang memberi dukungan dan mengingatkan bagaimana menjadi mahasiswa Batak perantau yang dapat diandalkan dan dibanggakan oleh sekitarnya.
9. Rekan-Rekan PMK ITS serta pak Kresnayana yang memberi dukungan dan mengingatkan untuk mengandalkan Tuhan disegala aktivitas kampus, karenanya penulis tidak lupa untuk berdoa dan mengucapkan syukur atas segala yang sudah dikerjakan, sehingga Tugas Akhir ini dapat dikerjakan dengan sukacita.
10. Rekan-rekan pelayanan Naposo se-Distrik Indonesia Bagian Timur, terutama Naposo Manyar yang selalu memberikan kehangatan kekeluargaan ditengah-tengah perantauan sehingga memberikan semangat kepada penulis.
11. Dimas, Alicia, Juli, Iza, Esis, Auliya, Aris, dan Delinda selaku kawan juang Tugas Akhir di Kedai 27 yang memberikan semangat dan sharing ilmunya dalam penyusunan Tugas Akhir ini.
12. Cynthia Siahaan yang sudah bersedia meluangkan waktu untuk berdiskusi mengenai penulisan Tugas Akhir ini.
13. Para warga Persatuan Statistika Jakarta (Jakarta Bersatu) yang selalu memberi semangat dan mengingatkan penulis dan membantu selama ini.

14. Seorang Dokter Muda di RSUD Soetomo yang tidak dapat disebutkan namanya, berkat diskusi dan tukar pikiran bersamanya, penulis memilih topik tugas akhir ini dan dapat mengetahui sudut pandang penelitian ini dari sisi Kedokteran.
15. Teman-teman Pejuang 116 atas semangat yang selalu diberikan kepada penulis.
16. Teman-teman Sigma 24 yang selalu memberikan kehangatan dan kenyamanan kepada penulis selama ini.
17. Semua pihak yang telah memberikan bantuan hingga penyusunan laporan Tugas Akhir ini dapat terselesaikan.

Penulis berharap hasil Tugas Akhir ini dapat bermanfaat bagi kita semua. Semoga kebaikan dan bantuan yang telah diberikan kepada penulis dibalas dengan kebaikan yang lebih besar lagi oleh Tuhan Yang Maha Esa. Amen.

Surabaya, Juni 2017

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
COVER PAGE	ii
LEMBAR PENGESAHAN	iv
ABSTRAK	vi
ABSTRACT	viii
KATA PENGANTAR	x
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xx
DAFTAR LAMPIRAN	xxi
BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	6
1.3 Tujuan Penelitian.....	6
1.4 Manfaat Penelitian.....	7
1.5 Batasan Penelitian	7
BAB II TINJAUAN PUSTAKA	
2.1 Magnetic Resonance Imaging (MRI)	9
2.2 Citra Digital.....	9
2.2.1 Digitalisasi Spasial	11
2.2.2 Citra abu-abu (<i>Grayscale</i>).....	12
2.3 <i>Operator sobel</i>	12
2.4 <i>Thresholding</i>	14
2.5 <i>Thresholding</i> metode Ridler-Calvard	14
2.6 Erosi	15
2.7 Dilasi	17
2.8 <i>Opening</i>	18
2.9 <i>Closing</i>	18
2.10 Segmentasi Berbasis Klaster	18
2.11 <i>Noise</i> Citra	19

2.12 <i>Gaussian Noise</i>	20
2.13 <i>Salt and Pepper Noise</i>	20
2.14 <i>Fuzzy C-Means</i>	21
2.15 <i>Expectation Maximization Algorithm</i>	24
2.16 <i>Expectation Maximization Algorithm</i> untuk klastering ...	25
2.17 Statistika Fitur Citra	27
2.18 <i>Gray Level Co-ocurrence Matrix (GLCM)</i>	28
2.19 <i>Energy (Angular Second Moment)</i>	31
2.20 <i>Correlation</i>	32
2.21 <i>Homogeneity</i>	32
2.22 <i>Contrast</i>	33
2.23 <i>Similarity</i>	32
2.24 <i>Uji Normalitas Kolmogorov Smirnov</i>	33
2.25 <i>Silhouette Index</i>	34
2.26 <i>Partition Coefficient Index</i>	35

BAB III METODOLOGI PENELITIAN

3.1 Sumber Data	37
3.2 Ruang Implementasi	37
3.2.1 Lingkungan Perangkat Keras.....	37
3.2.2 Lingkungan Perangkat Lunak.....	37
3.3 Variabel Penelitian dan Struktur data	37
3.4 Langkah Analisis	38
3.5 Diagram Alir.....	40

BAB IV IMPLEMENNTASI, UJI COBA, ANALISIS DAN PEMBAHASAN

4.1 Pemilihan data dan <i>proprocessing</i>	43
4.1.1 Data masukan <i>Preprocessing</i>	43
4.1.2 Histogram dan statistika fitur tekstur citra	51
4.2 Uji Coba, Pemilihan dan Validasi Jumlah klaster	67
4.2.1 Uji coba, Pemilihan dan Validasi klaster EM-GMM	67
4.2.2 Uji coba, Pemilihan dan Validasi klaster FCM	70
4.3 Segmentasi Citra dengan metode klastering	72
4.3.1 Segmentasi EM-GMM <i>ax T2 flair</i>	72
4.3.2 Segmentasi FCM <i>ax T2 flair</i>	79

4.4 Similarity	83
BAB V KESIMPULAN DAN SARAN	
5.1 Kesimpulan.....	87
5.2 Saran.....	87
DAFTAR PUSTAKA	89
LAMPIRAN	97
BIODATA PENULIS	133

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Digitalisasi Spasial.....	11
Gambar 2.2 Hubungan antara elemen gambar dan elemen <i>matrix</i>	11
Gambar 2.3 Perubahan Nilai <i>Pixel</i> Citra RGB ke dalam Citra Abu-Abu	12
Gambar 2.4 Citra Hasil <i>Thresholding</i>	13
Gambar 2.5 Densitas untuk mixture Normal.....	25
Gambar 2.6 Contoh Histogram Derajat Keabuan dan Jumlah Keabuan.....	28
Gambar 2.7 Ilustrasi hubungan tetangga derajat antar <i>pixel</i>	29
Gambar 2.8 Pembentukan <i>Co-ocurence Matrix</i>	30
Gambar 3.1 <i>Flowchart</i> langkah-langkah yang akan dilakukan dalam penelitian	41
Gambar 4.1 Data Citra masukan dan keluaran <i>preprocessing sequence ax T2 Flair</i>	44
Gambar 4.2 Citra Hasil setelah di <i>Thresholding</i> (a) Citra Awal (b) Citra Setelah di- <i>threshold</i>	45
Gambar 4.3 Citra Hasil <i>Head Contour</i> (a) Citra Asal <i>Thresholding</i> (b) Citra Setelah di tentukan <i>Head Contour</i>	46
Gambar 4.4 Citra Hasil <i>Head Masking</i>	46
Gambar 4.5 <i>Tim</i> Citra Hasil <i>Substraksi</i> dengan pengurangan <i>Head Mask</i> dengan citra <i>Threshold</i>	47
Gambar 4.6 Citra Hasil <i>Opening</i>	47
Gambar 4.7 Citra Hasil Proses Deteksi	48
Gambar 4.8 Citra Hasil Proses <i>Blocking</i>	48

Gambar 4.9	Citra Hasil <i>Substraksi</i> dengan pengurangan <i>blocking</i> dengan citra <i>opening</i>	49
Gambar 4.10	Citra Hasil Proses <i>Opening</i>	49
Gambar 4.11	Citra Hasil <i>preprocessing</i>	50
Gambar 4.12	Citra Hasil <i>preprocessing</i>	51
Gambar 4.13	Histogram Citra Sebelum <i>preprocessing</i>	51
Gambar 4.14	Histogram Citra Setelah <i>preprocessing</i>	52
Gambar 4.15	<i>Clustered Column Chart</i> selisih rata-rata Statistik Fitur Tekstur Citra sebelum dan setelah <i>preprocessing</i> data <i>ax T2 flair</i>	57
Gambar 4.16	Histogram Citra Sebelum <i>preprocessing</i>	60
Gambar 4.17	Citra Hasil <i>preprocessing</i>	61
Gambar 4.18	<i>Clustered Column Chart</i> selisih rata-rata Statistik Fitur Tekstur Citra sebelum dan setelah <i>preprocessing</i> data <i>ax T1 memp+c</i>	65
Gambar 4.19	Histogram Nilai <i>silhouette index ax T2 flair</i>	67
Gambar 4.20	Histogram Nilai <i>silhouette index ax T1 memp+c</i>	68
Gambar 4.21	Histogram Nilai <i>partition coefficient index ax T2 flair</i>	70
Gambar 4.22	Histogram Nilai <i>partition coefficient index ax T1 memp+c</i>	71
Gambar 4.23	Uji Coba Fungsi Keanggotaan <i>fuzzy c-means</i>	72
Gambar 4.24	Citra hasil Segmentasi 3 kluster	73
Gambar 4.25	Histogram Citra dan distribusi EM-GMM <i>ax T2 flair</i>	73
Gambar 4.26	Citra ROI dan Non-ROI segmentasi klustering EM-GMM	74
Gambar 4.27	<i>Subjective Analysis Region of Interest sequence ax T2 flair</i>	75
Gambar 4.28	Citra hasil Segmentasi 7 kluster	76

Gambar 4.29	Histogram Citra dan distribusi EM-GMM ax T2 memp+c	76
Gambar 4.30	Citra ROI dan Non-ROI segmentasi klastering EM-GMM	78
Gambar 4.31	<i>Subjective Analyst Region of Interest</i> <i>sequence ax T1 memp + C</i>	78
Gambar 4.32	Citra hasil Segmentasi 3 klaster.....	78
Gambar 4.33	Citra ROI dan Non-ROI segmentasi klastering FCM.....	79
Gambar 4.34	<i>Subjective Analyst Region of Interest</i> <i>sequence ax T2 flair</i>	80
Gambar 4.35	Citra hasil Segmentasi 6 klaster.....	81
Gambar 4.36	Citra ROI dan Non-ROI segmentasi klastering EM-GMM	81
Gambar 4.37	<i>Subjective Analyst Region of Interest</i> <i>sequence ax T1 memp+c</i>	82
Gambar 4.38	<i>Similarity</i> pada data ax T2 flair dengan <i>Gaussian Noise</i>	83
Gambar 4.39	<i>Similarity</i> pada data ax T2 flair dengan <i>Salt</i> <i>and Pepper Noise c</i>	84
Gambar 4.40	<i>Similarity</i> pada data ax T1 memp+c dengan <i>gaussian noise</i>	84
Gambar 4.41	<i>Similarity</i> pada data ax T1 memp+c dengan <i>Salt and Pepper Noise c</i>	85

DAFTAR TABEL

	Halaman
Tabel 3.1 Spesifikasi Perangkat Keras.....	37
Tabel 3.2 Variabel Penelitian.....	38
Tabel 3.3 Struktur Data	38
Tabel 4.1 Nilai Statistika Fitur Tekstur Citra Sebelum <i>preprocessing sequence ax T2 flair</i>	53
Tabel 4.2 Nilai Statistika Fitur Tekstur Citra Setelah <i>preprocessing</i>	56
Tabel 4.3 Nilai Statistika Fitur Tekstur Citra Sebelum <i>preprocessing</i>	61
Tabel 4.4 Nilai Statistika Fitur Tekstur Citra Setelah <i>preprocessing</i>	64
Tabel 4.5 Uji Kolmogorov Smirnov data tiap-tiap kelas <i>ax T2 Flair</i>	69
Tabel 4.6 Uji Kolmogorov Smirnov data tiap-tiap kelas <i>ax T1 Memp+c</i>	69
Tabel 4.7 Nilai Pusat klaster FCM <i>sequence ax T2 flair</i>	80
Tabel 4.8 Nilai Pusat klaster FCM <i>sequence ax T2 flair</i>	82

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 syntax preprocessing main.m	97
Lampiran 2 syntax preprocessing thresholding.m	97
Lampiran 3 syntax preprocessing contourandmask.m.....	98
Lampiran 4 syntax preprocessing brainarea.m	99
Lampiran 5 syntax menghitung nilai statistika fitur citratekstureanalysis.m.....	101
Lampiran 6 syntax pemberian noise pada citra noise.m	104
Lampiran 7 syntax similarity.m.....	105
Lampiran 8 syntax menghitung nilai silhouette silhouette.m	105
Lampiran 9 syntax klustering EM-GMM klaster emgmmklaster.m.....	105
Lampiran 10 Syntax klustering FCM klaster fcmklaster.m	110
Lampiran 11 Hasil Segmentasi Citra <i>ax T2 flair</i> metode EM- GMM.....	115
Lampiran 12 Hasil Segmentasi Citra <i>ax T2 flair</i> metode FCM	118
Lampiran 13 Hasil Segmentasi Citra <i>ax T1 memp+c</i> metode EM-GMM	121
Lampiran 14 Hasil Segmentasi Citra <i>ax T1 memp+c</i> metode FCM	124
Lampiran 15 Nilai <i>silhouette index</i>	127
Lampiran 16 Nilai <i>partition coefficient index</i>	127

Lampiran 17	Nilai <i>similarity</i> data <i>ax</i> T2 <i>flair</i> terhadap <i>Gaussian noise</i>	128
Lampiran 18	Nilai <i>similarity</i> data <i>ax</i> T1 <i>memp+c</i> terhadap <i>Gaussian noise</i>	128
Lampiran 19	Nilai <i>similarity</i> data <i>ax</i> T2 <i>flair</i> terhadap <i>salt</i> <i>and pepper noise</i>	128
Lampiran 20	Nilai <i>similarity</i> data <i>ax</i> T1 <i>memp+c</i> terhadap <i>salt and pepper noise</i>	129

BAB I

PENDAHULUAN

1.1 Latar Belakang

Otak merupakan bagian yang paling penting dari tubuh manusia dimana segala aktivitas yang dilakukan manusia pada tiap harinya dikendalikan oleh otak. Otak merupakan syaraf pusat yang mengkoordinir dan mengatur seluruh tubuh dan pemikiran manusia, oleh karna itu, perlakuan khusus harus dilakukan terhadap otak, sehingga di segala aktivitas pada bagian kepala manusia harus memerlukan perhatian khusus, karena jika terjadi cedera kepala sedikit saja dapat mengakibatkan malapetaka besar bagi seseorang. Kasus yang paling mematikan dan sering terjadi pada otak adalah tumor otak. Tumor merupakan pertumbuhan sel yang tidak normal dalam tubuh, tidak banyak penderita yang dapat bertahan hidup, terutama terhadap penyakit tumor otak yang ganas kasus selama 20 tahun terakhir ini. Tumor otak juga menjadi penyebab terjadinya penyakit lainnya, seperti tumor syaraf pendengaran (*neurilemoma*), dan tumor pembuluh darah (Price dan Lorraine, 2006).

Berdasarkan data-data “*Surveillance Epidemiology & End Result Registry USA*” dari tahun 1973-1995 dilaporkan bahwa setiap tahunnya di USA dijumpai 38.000 kasus baru tumor otak primer, dan pada tahun 2001 dijumpai lebih dari 180.000 kasus tumor otak, (Soffieti, 2003). Insiden tumor otak lebih sering dijumpai pada laki-laki (6,3 dari 100.000 penduduk) dibanding perempuan (4,4 dari 100.000 penduduk), dengan kelompok usia terbanyak sekitar 65 – 79 tahun (Hakim, 2006). Untuk mengetahui keberadaan penyakit tumor tersebut, dilakukan deteksi area tumor. Deteksi area tumor otak sering kali dilakukan oleh para pekerja medis, seperti Dokter dan Ahli Radiologi, dalam hal pendeteksian ini menggunakan data citra dalam menentukan lokasi tumor otak tersebut (Malik, 2009).

Dalam melakukan analisis citra medis sudah banyak dikembangkan alat-alat, salah satunya adalah MRI (*Magnetic*

Resonance Imaging) yang merupakan alat yang menggunakan teknik pencitraan diagnostik yang sangat bermanfaat dalam mendeteksi dini perubahan abnormal pada jaringan dan organ yang tidak melibatkan paparan radiasi seperti *X-ray*, sehingga teknik ini dapat memberikan informasi yang sangat jelas yaitu berupa visualisasi struktur anatomi tubuh manusia (Nurhayati, 2002). MRI juga sudah kerap dilakukan oleh para peneliti didalam prosesi gambar medis. MRI juga merupakan salah satu teknik yang penting untuk mendeteksi dini penyakit tumor. Dalam diagnosanya, MRI dipakai di Rumah Sakit Umum Daerah (RSUD) Soetomo di Surabaya. Alat yang dipakai di RSUD Soetomo memakai MRI berkekuatan 1,5 Tesla pada perkembangan alatnya sudah ditemukan alat MRI yang lebih baru yaitu MRI berkekuatan 3 tesla dan belum digunakan di RSUD Soetomo. Alat MRI berkekuatan 3 Tesla tersebut memberikan hasil penggambaran citra yang lebih baik daripada MRI 1,5 Tesla dan sudah banyak di pakai di Rumah Sakit swasta di Surabaya.

Parameter data MRI dapat menerima beragam tingkat keabu-abuan (*grayscale*) dan beragam bentuk data organ tubuh manusia (Tian dan Fan, 2007), sehingga dalam pemrosesan datanya dilakukan proses segmentasi gambar berdasarkan data MRI tersebut. Dalam proses segmentasi citra MRI sering kali ditemukan kesalahan hasil segmentasi dan membutuhkan waktu yang lama untuk membedakan area tumornya. Hal ini terjadi karena keterbatasan dan kesalahan manusia dalam membaca citra digitalnya, oleh sebab itu dibutuhkan metode analisis lebih lanjut untuk segmentasi data citra MRI tersebut, yang dapat memperkecil kesalahan dalam membaca data citra. Data citra MRI merupakan suatu *Grayscale Image* yang dapat direpresentasikan dalam bentuk *array* dua dimensi, dan tiap-tiap elemen pada *array* tersebut menunjukkan intensitas (*Grayscale*) dari image pada posisi yang bersesuaian. Intensitas *grayscale* tersebut merupakan bahan dasar dilakukannya klastering (Nurhayati, 2002).

Segmentasi gambar citra MRI populer dilakukan dengan menggunakan algoritma klastering. Proses kerja algoritma

klustering dalam segmentasi citra adalah membagi citra menjadi beberapa bagian tergantung jumlah kluster yang akan digunakan. Algoritma klustering terbagi menjadi dua buah jenis algoritma, yaitu: *hard clustering* dan *soft clustering*. Algoritma *hard clustering* merupakan algoritma klustering yang membagi anggotanya berdasarkan jaraknya terdekat terhadap pusat kluster yang ada dan secara tegas suatu anggota akan dinyatakan benar atau tidaknya menempati suatu kluster. Lainnya, pada algoritma *soft clustering* pembagian anggota klasternya berdasarkan besarnya derajat keanggotaan dari suatu kluster yang bernilai diantara 0 sampai 1.

Untuk data citra MRI ini merupakan sebuah intensitas *grayscale* di tiap-tiap *pixel*-nya. Saat dilakukan implementasi klustering untuk segmentasi terhadap data citra ini terjadi permasalahan yaitu intensitas *pixel-pixel* yang ada, akan menarik *pixel-pixel* lainnya yang memiliki intensitas yang hampir sama ke dalam suatu kluster yang sama, sehingga tidak *robust* terhadap *noise* (Shan, 2005). Dengan permasalahan tidak *robust*-nya klustering terhadap *noise*, saat melakukan klustering citra, algoritma *hard clustering* akan mengalami kegagalan juga untuk *converge*, sebab metode *hard clustering* mengalokasikan secara tegas (benar atau tidak) suatu anggota menjadi suatu kluster tertentu, jika terjadi perpindahan suatu data akibat *noise* ke dalam suatu kluster tertentu dapat mengubah model klasternya. Akibatnya, pemodelan klustering yang terbaru tidak memberikan hasil yang lebih baik sebelum data tersebut dipindahkan, sehingga algoritma *hard clustering* mengalami kegagalan untuk *converge*. Permasalahan ini sangat kecil terjadi jika menerapkan algoritma *soft clustering*, sebab *pixel-pixel*-nya akan diperlengkapi dengan derajat keanggotaan, sehingga lebih *robust* saat diimplementasikan ke dalam segmentasi citra (Pena, Lozano, Larragana, 1999).

Metode *soft clustering* dapat diimplementasikan lebih baik untuk data segmentasi citra. Metode *soft clustering* dalam pendekatannya dapat dibedakan menjadi dua jenis, yaitu: metode

yang *soft clustering* yang mengikuti bentuk pola distribusi dan *soft clustering* yang tidak mengikuti bentuk pola distribusi. Untuk *soft clustering* yang mengikuti bentuk pola distribusi contohnya adalah *Mixture Model*. Metode tersebut memodelkan banyaknya intensitas *grayscale* tiap level intensitasnya membentuk beberapa kelompok distribusi sehingga tiap-tiap kelompok distribusi tersebut di representasikan sebagai sebuah kluster (Balafar, 2013). Pada umumnya jenis distribusi yang dipakai dalam untuk klastering adalah distribusi *Gaussian* atau sering dikenal sebagai distribusi Normal. Segmentasi sudah pernah dilakukan menggunakan metode *Gaussian Mixture Model* (GMM) didapatkan hasil yang kurang baik, sebab data intensitas ini tidak mengikuti distribusi Normal (Zarpak dan Rahman, 2008). Mengatasi permasalahan tersebut, GMM untuk segmentasi citra harus dioptimasi sehingga mendapatkan hasil yang lebih baik. Metode optimasi yang digunakan adalah *Expectation Maximization* (EM). Dalam penelitian Rohani tahun 2013, metode EM-GMM ini memberikan hasil rata-rata akurasi yang cukup tinggi yaitu sebesar 80% dalam segmentasi citra *CT Scan*.

Metode *soft clustering* lainnya adalah metode *soft clustering* yang tidak mengikuti pola distribusi data. Dalam segmentasi citra, metode yang paling sering adalah *Fuzzy C means* untuk *soft clustering* yang tidak mengikuti pola distribusi. Penelitian sebelumnya adalah penelitian yang dilakukan oleh Bhagwat, More, Shinde, Daga, dan Tornekar (2013), Citra MRI tersebut dikomparasi menggunakan metode *K-Means*, *Fuzzy C-Means*, dan *Hierarchical Cluster*. Hasil yang diperoleh menyatakan bahwa untuk klastering menggunakan metode *Fuzzy C-Means* memberikan nilai *similarity* sebesar 70% untuk data bukan medis dan 83,33% untuk data medis.

Metode klastering merupakan pemrosesan data secara alami dengan algoritma yang berjalan sendiri sehingga didapatkan kelompok-kelompok yang terbentuk secara alami atau sering juga disebut sebagai *unsupervised*. Penentuan jumlah kluster sering kali terjadi saat melakukan proses klastering sehingga diperlukan

evaluasi pada sistem tersebut. Evaluasi sistem merupakan bagian penting dalam penentuan jumlah kluster yang akan dipakai. Pada dasarnya penentuan jumlah kluster untuk segmentasi citra MRI ditentukan oleh pendapat subjektif dari pakar medis. Penentuan bagian-bagian area otak dan area tumor otak ditentukan hanya secara pengamatan saja. Tetapi pada saat dilakukan segmentasi tersebut diperlukan analisis terhadap hasil eksternalnya diperlukan evaluasi (Tan, Steinbach, dan Kumar, 2006).

Evaluasi validasi sistem kluster dapat dilakukan dengan menggunakan nilai *silhouette index* dan *Partition Coefficient Index* (PCI). Bezdek (1981) mengusulkan validitas dengan menghitung koefisien partisi atau *partition coefficient index* (PCI) sebagai evaluasi nilai keanggotaan setiap kluster untuk metode FCM. Sedangkan, untuk metode EMGMM tidak dapat menerapkan nilai PCI sebab tidak mempunyai derajat keanggotaan. Hal ini dapat diatasi menggunakan nilai SI. Penerapan nilai SI terhadap klustering adalah membandingkan rata-rata jarak suatu data dan data lainnya dalam sebuah kluster dengan jarak sebuah data dan data lainnya yang berbeda kluster. Sehingga akan dilakukan validasi jumlah kluster menggunakan PCI dan SI. Nilai-nilai medis ini yang menjadi pertimbangan dalam memilih kluster terbaik, tetapi pemilihan dengan pendapat medis menjadi pertimbangan yang lebih utama.

Berdasarkan pemaparan di atas dalam penelitian ini akan dilakukan kluster citra MRI, yang akan dilakukan *preprocessing* terlebih dahulu, lalu menghitung nilai Statistika Fitur Tekstur Citra *Gray Level Co-ocurrence Matrix* (GLCM) berdasarkan elemen-elemen dari GLCM akan dihitung nilai *Energy*, *Correlation*, *Homogeneity*, dan *Contrast*. Citra tersebut akan diubah menjadi sebuah *matrix*, dimana tiap-tiap elemen dari *matrix* tersebut berupa nilai-nilai intensitas keabuan (*grayscale*). Berdasarkan *matrix* tersebut akan dikomparasi menggunakan metode kluster *Fuzzy C-Means* (FCM) dan *Gaussian Mixture Model* (GMM) dengan optimasi *Expectation Maximization* (EM) atau dapat disebut juga dengan EM-GMM. Pemilihan jumlah kluster akan ditentukan

menggunakan nilai *silhouette index* (SI) dan *partition coefficient index* (PCI). Lalu, dengan menggunakan kedua metode tersebut, akan dibandingkan nilai *similarity*, dimana citra MRI tersebut akan diberikan *noise* menggunakan metode *Gaussian Noise* dan *Pepper and Salt Noise* dengan tingkat 1%, 5%, dan 10%. Lalu, citra yang sudah diberi *noise* akan disegmentasi menggunakan metode FCM dan EM-GMM, berdasarkan perubahan klastering tiap-tiap *pixel* akan dihitung total nilai *similarity*nya. Untuk mempermudah melakukan penelitian ini, akan dibuat program menggunakan *Integrated Development Environment* (IDE) Matlab.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah diuraikan diatas, maka dapat dirumuskan masalah-masalah diantara lain adalah suatu citra MRI yang direpresentasikan menjadi sebuah *matrix* yang memiliki intensitas *grayscale*. Area tumor dan tulang otak memiliki intensitas keabuan yang sama sehingga harus dilakukan *preprocessing*, sehingga harus dilakukan pengukuran kualitas citra sebelum dan sesudah dilakukan *preprocessing*, dengan menggunakan Statistika Fitur Tekstur Citra. Dalam mendapatkan visualisasi area tumor otak, citra MRI tersebut akan disegmentasi dengan metode klastering, yaitu FCM dan EM-GMM sebelumnya akan dilakukan pemilihan jumlah klaster menggunakan nilai SI untuk EM-GMM dan PCI untuk FCM sebagai pertimbangan awal pemilihan jumlah klaster tetapi pendapat subjektif medis menjadi pilihan utama. Metode klastering yang dipakai akan dikomparasi menggunakan nilai *similarity* untuk menunjukkan dari kedua metode yang digunakan metode yang mana lebih robust terhadap *noise* dengan tingkat 1%, 5%, 10%..

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dibuat, dapat diperoleh tujuan utama sebagai berikut:

1. Mendapatkan perbandingan nilai Statistika Fitur Tekstur Citra sebelum dan setelah melakukan *preprocessing*.

2. Memperoleh jumlah klaster terbaik untuk masing-masing metode berdasarkan nilai SI dan PCI. Pemilihan jumlah klaster selain didasari nilai SI dan PCI juga berdasarkan pertimbangan analisa subjektif berdasarkan medis.
3. Memperoleh hasil klastering FCM untuk segmentasi citra MRI.
4. Memperoleh hasil klastering EM-GMM untuk segmentasi citra MRI.
5. Memperoleh hasil *similarity* FCM dan EM-GMM pada segmentasi citra MRI dengan tingkat *noise* 1%, 5% dan 10%.

1.4 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah memberikan hasil segmentasi citra MRI sebaik bahkan lebih baik dari alat MRI 3 Tesla.

1.5 Batasan Masalah

Beberapa batasan yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Lingkungan implementasi menggunakan IDE Matlab.
2. Nilai Statistika Fitur Tekstur Citra digunakan untuk membedakan kualitas gambar sebelum dan sesudah dilakukannya *preprocessing*.
3. Nilai m untuk FCM yang dipakai adalah 2, untuk mengantisipasi fungsi objektif menghasilkan nilai optimum lebih dari satu.
4. Pengujian klastering berdasarkan nilai Validasi indeks dan diikuti analisa subjektif dari pihak medis.
5. Data yang akan diujikan adalah data satu pasien berdasarkan satu *sequence*/bagian otak citra MRI yang diperoleh dari Rumah Sakit Umum Daerah Soetomo.
6. Output yang dihasilkan adalah berupa citra yang sudah tersegmentasi berdasarkan metode klastering dan dapat ditentukan *Region of Interest* dan *Non Region Interestnya*

7. Hasil segmentasi dengan algoritma FCM dan EM-GMM akan dikomparasi tingkat *similarity*-nya.

BAB II

TINJAUAN PUSTAKA

2.1 Magnetic Resonance Imaging (MRI)

Prinsip dasar MRI adalah inti atom yang bergetar dalam medan magnet. Proton yang merupakan inti atom hidrogen yang bila ditembak tegak lurus pada intinya dengan medan berfrekuensi tinggi secara periodik, maka proton tersebut akan bergentar. Alat ini digunakan pada tubuh manusia karena manusia mempunyai konsentrasi atom hidrogen yang tinggi (70%) dan bila medan magnet berfrekuensi tinggi ini dimatikan maka proton yang bergetar tadi akan kembali keposisi semula dan akan menginduksi suatu kumparan untuk menghasilkan sinyal elektrik yang lemah. Bila hal ini terjadi berulang-ulang dan sinyal elektrik tersebut ditangkap kemudian diproses dalam komputer akan dapat disusun suatu gambar (lohmann, 1998).

MRI adalah suatu teknik pencitraan medis dalam pemeriksaan diagnostik radiologi, yang menghasilkan rekaman citra potongan penampang tubuh atau organ manusia dengan menggunakan medan magnet dan resonansi getaran terhadap inti atom hidrogen. Dapat juga dikatakan bahwa MRI merupakan alat diagnostik *imaging* atau alat pemeriksaan radiologi berteknologi tinggi, yang menggunakan medan magnet yang besar, frekuensi radio tertentu dan seperangkat detektor dan pengolahan data untuk menghasilkan gambaran potongan-potongan anatomi tubuh manusia. Dengan pencitraan MRI dapat dihasilkan potongan atau irisan (*slice*) melintang atau tegak sesuai dengan kebutuhan (lohmann, 1998). Data citra MRI yang dipakai dalam penelitian ini merupakan citra yang diambil menggunakan MRI berkekuatan sebesar 1.5 T. Alat tersebut dapat mengambil dapat menghasilkan citra yang memiliki ukuran *pixel* 256x256 dan 512x512.

2.2 Citra Digital

Citra sebagai keluaran suatu sistem pengambilan atau perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor TV. Proses

pengolahan citra dapat dikatakan sebagai *image processing*. *Image processing* merupakan proses yang berhubungan dengan cara untuk meningkatkan kualitas suatu image sehingga representasi image tersebut sesuai dengan kemampuan manusia. Sedangkan *machine vision* berhubungan dengan cara memproses suatu image sehingga didapat data-data image yang dapat dimengerti oleh komputer untuk analisa lebih lanjut. Kedua proses ini sangat penting dalam *image/patern recognition* (Xu dan Donald, 2009).

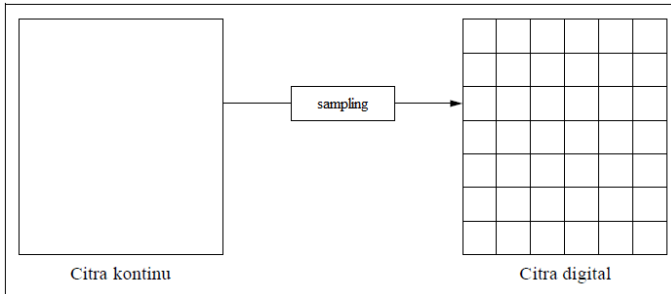
Secara digital suatu *grayscale* dapat direpresntasikan dalam bentuk *array* dua dimensi. Tiap elemen pada *array* tersebut menunjukkan intensitas (*grayscale*) dari image pada posisi koordinat yang bersesuaian. Apabila suatu image direpresentasikan dalam 8 bit, pada image terdapat 2^8 atau 256 level grayscale, biasanya bernilai 0-255 (Gonzalez dan Woods, 2002). Dimana 0 menunjukkan level intensitas yang paling gelap dan 255 menunjukkan intensitas paling terang. Tiap elemen pada *array* diatas disebut sebagai *picture element* atau sering disebut sebagai *pixel*. Dengan melakukan perubahan pada intensitas masing-masing *pixel* maka representasi image secara keseluruhan akan berubah. Image yang dinyatakan dengan $M \times N$ *matrix* mempunyai intensitas tertentu pada *pixel* tertentu. Posisi picture elemen (i, j) dan koordinat *pixel* (x, y) berbeda (Xu dan Donald, 2009), Representasi citra dalam *matrix* dapat ditunjukkan pada persamaan (2.1), dimana setiap $f(x, y)$ merupakan tingkat intensitas atau tingkat keabu-abuan (*grayscale*) dari citra tiap-tiap lokasi *pixel*nya (x, y).

$$f(i, j) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, M-1) \\ f(1,0) & \dots & \dots & f(1, M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1, M-1) \end{bmatrix} \quad (2.1)$$

Indeks baris (i) dan indeks kolom (j) menyatakan suatu koordinat titik pada citra, sedangkan $f(i, j)$ merupakan intensitas (derajat keabuan) pada titik (i, j) (Gonzalez dan Woods, 2002).

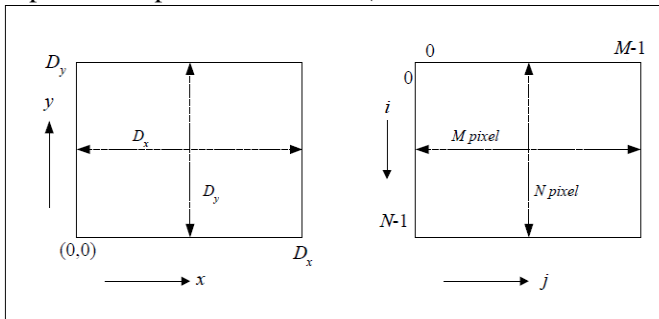
2.2.1 Digitalisasi Spasial

Digitalisasi spasial (x, y) sering disebut sebagai sampling. Sampling dalam hal ini merupakan membagi-bagi citra kedalam grid berdasarkan ukuran *pixel*-nya. Dalam proses ini 1 *pixel* akan mewakili 1 grid (Gonzalez dan Woods, 2002). Citra kontinu disampling pada grid-grid yang berbentuk bujur sangkar dapat dilihat pada Gambar 2.1



Gambar 2.1 Digitalisasi Spasial

Terdapat perbedaan antara koordinat gambar (yang disampling) dengan koordinat matriks (hasil digitalisasi). Titik asal $(0, 0)$ pada gambar dan elemen $(0, 0)$ pada *matrix* tidak sama. Koordinat x dan y pada gambar dimulai dari sudut kiri bawah, sedangkan penomoran *pixel* pada *matrix* dimulai dari sudut kiri atas dapat dilihat pada Gambar 2.2 (Gonzalez dan Woods, 2002).



Gambar 2.2 Hubungan antara elemen gambar dan elemen *matrix*

Keterangan gambar:

$$i = (N-1-y), \quad 0 \leq i \leq N-1,$$

- $j = x, 0 \leq j \leq M-1,$
 $x = D_x / M$, pergerakan jarak pixel dari pusat ke arah vertikal,
 $y = D_y / N$, pergerakan jarak pixel dari pusat ke arah horizontal,
 N = jumlah maksimum *pixel* dalam satu baris,
 M = jumlah maksimum *pixel* dalam satu kolom,
 D_x = lebar gambar (dalam inchi),
 D_y = tinggi gambar (dalam inchi).

Tiap-tiap elemen (i,j) di dalam matriks menyatakan nilai intensitas abu-abu.

2.2.2 Citra Abu-Abu (*Grayscale*)

Citra abu-abu merupakan citra yang hanya memiliki satu nilai pada setiap *pixel*nya, nilai tersebut disebut dengan kadar keabuan (Gonzalez dan Woods, 2002). Citra abu-abu memiliki komposisi *Red* (R), *Green* (G), dan *Blue* (B). Proses perubahan citra berwarna ke dalam citra abu-abu dapat dilakukan dengan pembobotan. Pembobotan ini didasari pada tingkat kepekaan manusia dalam memandang komposisi warna. Persamaan metode pembobotan ini dapat dirumuskan pada persamaan (2.2) (Shih, 2010).

$$gray = 0.299R + 0.578G + 0.114B \quad (2.2)$$

Dimana berdasarkan persamaan (2) dapat dilakukan perubahan nilai komposisi RGB ke dalam citra abu-abu, seperti pada Gambar 2.3

R=200 G=0 B=100	R=50 G=50 B=80	R=140 G=0 B=100
R=200 G=200 B=20	R=80 G=80 B=95	R=20 G=0 B=10
R=10 G=0 B=20	R=200 G=0 B=100	R=0 G=0 B=0

71	52	53
198	80	7
5	71	0

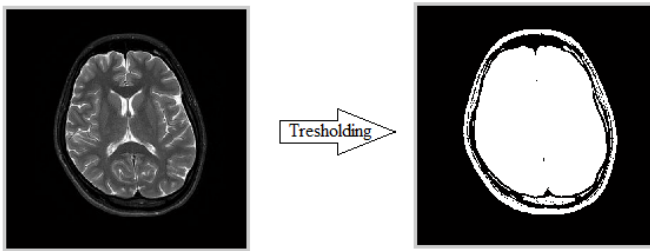
Gambar 2.3 Perubahan Nilai *Pixel* Citra RGB ke dalam Citra Abu-Abu

2.3 Thresholding

Thresholding merupakan bagian dari *preprocessing*. Secara umum proses *thresholding* terhadap citra *grayscale* bertujuan menghasilkan citra biner, secara matematis dapat ditulis didalam persamaan (2.9) (Prasetyo, 2011).

$$g(x, y) = \begin{cases} 1, & \text{jika } f(x, y) \geq T \\ 0, & \text{jika } f(x, y) < T \end{cases} \quad (2.9)$$

Dimana $g(x,y)$ adalah citra biner dari citra *grayscale* $f(x,y)$ dan T menyatakan nilai *threshold*, (Prasetyo, 2011) ilustrasi *thresholding* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Citra hasil *Thresholding*

Berdasarkan proses *thresholding* pada Gambar 2.4, citra awal yang memiliki intensitas abu-abu diubah menjadi citra biner berdasarkan tingkat *thresholding*-nya.

2.4 Thresholding Metode Ridler-Calvard

Salah satu proses penting dalam *preprocessing*. Salah satu metode *thresholding* paling sederhana adalah *thresholding* Metode Ridler-Calvard akan bekerja dengan baik jika histogram citra bukan merupakan *bi-modal* (Sonka, Hlavac, dan Boyle, 2008). metode Ridler-Calvard pertama kali dipublikasikan pada Agustus 1978. Langkah-langkah Metode Ridler-Calvard adalah sebagai berikut:

1. Tentukan Nilai *threshold* pertama

Nilai *threshold* pertama didapat berdasarkan persamaan (2.10).

$$T_0 = \frac{\sum_{i=0}^n x_i}{n} \quad (2.10)$$

Dimana,

T_0 = nilai *threshold* pertama

x_i = nilai intensitas *pixel*

n = jumlah data, nilai n berupa bilangan bulat

2. Kelompokkan data citra masukan

Setelah nilai *threshold* didapat, nilai intensitas *pixel* yang lebih dari nilai *threshold* akan dikelompokkan dalam variabel tersendiri. Begitu juga nilai intensitas *pixel* kurang dari sama dengan nilai *threshold*. Pengelompokan data didasarkan pada persamaan (2.11).

$$\begin{aligned} x_i &\in G, \text{ jika } x_i \geq T_0 \\ x_i &\in L, \text{ jika } x_i < T_0 \end{aligned} \quad (2.11)$$

Dimana:

x_i = nilai intensitas *pixel*

G = himpunan *pixel* yang nilainya lebih dari sama dengan *threshold*

L = himpunan *pixel* yang nilainya kurang dari *threshold*

3. Tentukan nilai *threshold* baru

Untuk menentukan nilai *threshold* baru, diperlukan jumlah nilai rata-rata dari G dan L . Jumlah nilai rata-rata dari G dan L didapat berdasarkan persamaan (2.12).

$$\text{jumlah} = \frac{\sum_{i=0}^n G_i}{n} + \frac{\sum_{i=0}^n L_i}{n} \quad (2.12)$$

Dimana:

G_i = Nilai intensitas G ke- i

L_i = Nilai intensitas L ke- i

n = jumlah data

2.3 Operator Sobel

Operator Sobel merupakan bagian dari *preprocessing*. *Operator Sobel* merupakan algoritma yang digunakan untuk mendeteksi tepian gambar. Masing-masing *pixel* pada citra input akan dikonvolusi menggunakan *matrix* berukuran 3 x 3. Konvolusi adalah proses mengubah nilai tiap-tiap *pixel* menggunakan suatu fungsi yang melibatkan nilai *pixel* tetangganya (Prasetyo, 2011).

Dengan memisalkan $f(1,1)$ dalam persamaan (2.3) adalah pixel yang akan di konvolusi. Tiap-tiap *pixel* representasi citra pada persamaan (2.3) akan dikalikan dengan *operator sobel*, baik secara vertikal menggunakan persamaan (2.4), dan secara horizontal atau menggunakan persamaan (2.5) (Prasetyo, 2011).

$$\begin{pmatrix} f(0,0) & f(0,1) & f(0,2) \\ f(1,0) & f(1,1) & f(1,2) \\ f(2,0) & f(2,1) & f(2,2) \end{pmatrix} \quad (2.3)$$

$$\text{operator sobel horizontal} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad (2.4)$$

$$\text{operator sobel vertikal} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \quad (2.5)$$

Persamaan Gradien Sobel untuk horizontal ditunjukkan oleh persamaan (2.6), dan Persamaan Gradien Sobel untuk vertikal ditunjukkan oleh persamaan (2.7)

$$\begin{aligned} f(1,1) &= (f(2,0) + 2f(2,1) + f(2,2)) \\ &\quad - (f(0,0) + 2f(0,1) + f(0,2)) \end{aligned} \quad (2.6)$$

$$\begin{aligned} f(1,1) &= (f(0,2) + 2f(1,2) + f(2,2)) \\ &\quad - (f(0,0) + 2f(1,0) + f(2,0)) \end{aligned} \quad (2.7)$$

Proses tersebut akan dilakukan secara horizontal terlebih dahulu dari kiri sampai kanan atau dari j sampai $M-1$ untuk semua element i . Setelah itu proses berlanjut secara vertikal dari atas sampai bawah atau dari i sampai $N-1$ untuk semua element j .

2.6 Erosi

Erosi merupakan bagian dari *preprocessing*. Erosi merupakan proses penghapusan titik-titik batas objek menjadi bagian dari latar, berdasarkan *structuring element* yang digunakan. Pada operasi ini, ukuran obyek diperkecil dengan mengikis

sekeliling objek. Ada 2 cara yang dapat dilakukan untuk proses erosi, yaitu :

1. Dengan mengubah semua titik batas menjadi titik latar
2. Dengan menset semua titik di sekeliling titik latar menjadi titik latar

Jika diketahui himpunan A dan B , maka erosi A oleh B (dinotasikan $A \ominus B$) didefinisikan didalam persamaan (2.13) (Gonzalez dan Woods, 2002).

$$A \ominus B = \{w: B_w \subseteq A\} \quad (2.13)$$

Dengan kata lain, erosi A oleh B terdiri atas semua titik $w = (x, y)$ dimana B_w ada di dalam himpunan A . Untuk melakukan erosi, B digeser-geser dalam A dan dicari dimana saja B bebarbenar ada di dalam A . Untuk kondisi-kondisi yang memenuhi syarat tersebut maka tandailah titik (0,0) yang bersesuaian dengan B . Titik-titik inilah yang merupakan hasil erosi A oleh B (Gonzalez dan Woods, 2002).

2.7 Dilasi

Dilasi merupakan bagian dari *preprocessing*. Dilasi merupakan proses penggabungan titik-titik latar menjadi bagian dari objek, berdasarkan *structuring element* yang digunakan. Proses ini adalah kebalikan dari erosi, yaitu merubah latar disekeliling objek menjadi bagian dari objek tersebut. Ada 2 cara untuk melakukan operasi ini, yaitu (Prasetyo, 2011):

1. Dengan cara mengubah semua titik latar yang bertetangga dengan titik batas menjadi titik obyek, atau lebih mudahnya set setiap titik yang tetangganya adalah titik obyek menjadi titik obyek.
2. Dengan mengubah semua titik di sekeliling titik batas menjadi titik obyek, atau lebih mudahnya set semua titik tetangga sebuah titik obyek menjadi titik obyek.

Misalkan A dan B adalah himpunan-himpunan piksel. Dilasi A oleh B dinotasikan dengan $A \oplus B$ dan didefinisikan di dalam persamaan (2.14).

$$A \oplus B = \bigcup_{x \in B} A_x \quad (2.14)$$

Ini berarti bahwa untuk setiap titik $x \in B$, maka dilakukan translasi atau pergeseran dan kemudian menggabungkan seluruh hasilnya (union) (Prasetyo, 2011).

2.8 Opening

Opening merupakan bagian dari *preprocessing*. *Opening* merupakan proses morfologi yang menggabungkan proses dilasi dan erosi. Pada mulanya, citra dierosi kemudian didiasi. Tujuan *opening* adalah menghilangkan objek-objek yang ukurannya lebih kecil dari *structure element* dan memisahkan objek yang semula bergandengan (Gonzalez dan Woods, 2002). *Opening* dapat didefinisikan di dalam persamaan (2.15).

$$A \circ B = (A \ominus B) \oplus B \quad (2.15)$$

Sehingga operasi *opening* merupakan sebuah operasi yang terdiri atas operasi erosi diikuti oleh operasi dilasi. Definisi ekivalennya dapat dinyatakan didalam persamaan (2.16) (Gonzalez dan Woods, 2002).

$$A \circ B = \bigcup \{B_w : B_w \subseteq A\} \quad (2.16)$$

Dimana $A \circ B$ merupakan abungan (*union*) dari seluruh pergeseran B yang benar-benar tercakup (*fit in*) dalam A .

2.9 Closing

Closing merupakan bagian dari *preprocessing*. *Closing* didefinisikan sebagai operasi dilasi yang dilanjutkan dengan operasi erosi, dinotasikan sebagai $A \bullet B$, sehingga dapat dinyatakan didalam persamaan (2.17) (Gonzalez dan Woods, 2002).

$$A \bullet B = (A \oplus B) \ominus B \quad (2.17)$$

Tujuan dari morfologi *closing* adalah mengisi lubang-lubang kecil pada objek dan menggabungkan objek yang berdekatan.

2.10 Segmentasi Berbasis Klaster

Segmentasi merupakan seni atau teknik mengelompokkan data (objek), kedalam beberapa klaster (kelompok) yang belum diketahui. Klastering adalah metode penganalisaan data, yang

sering dimasukkan sebagai salah satu metode *Data Mining*, yang tujuannya adalah untuk mengelompokkan data dengan karakteristik yang sama pada suatu area yang sama dan data dengan karakteristik yang berbeda ke area lainnya. Berbeda dengan metode klasifikasi, pada teknik klastering, jumlah dan karakteristik dari kelompok akan diperoleh dari data kelompok, biasanya belum diketahui sebelumnya. Secara umum, metode klastering dikelompokkan menjadi 2 yaitu metode berbasis hirarki (*hierarchical*) dan metode berbasis partisi (*non-hierarchical*) (Xu dan Donald, 2009).

Metode segmentasi berbasis *cluster* merupakan salah satu bagian penting dari pemrosesan citra yang bertujuan untuk pengelompokkan data atau penggolongan objek berdasarkan informasi yang diperoleh dalam kelas-kelas, menjadi beberapa region yang homogen berdasarkan kriteria kemiripan tertentu, sehingga data dalam suatu *cluster* memiliki tingkat persamaan yang tinggi satu dengan lainnya, tetapi sangat berbeda dengan data *cluster* lain (Xu dan Donald, 2009).

2.11 Noise Citra

Fokus penelitian ini adalah membandingkan ke-*robust*-an metode klaster terhadap *noise* citra. *Noise* Citra muncul biasanya sebagai akibat dari pembelokan cahaya yang tidak baik. Gangguan tersebut umumnya berupa variasi intensitas suatu *pixel* yang tidak berkorelasi dengan *pixel-pixel* tetangganya. Secara visual, gangguan mudah dilihat oleh mata karena tampak berbeda dengan piksel tetangganya. *Pixel* yang mengalami gangguan umumnya memiliki frekuensi tinggi (Munir, 2004).

Noise dapat disebabkan oleh gangguan fisik (optik) pada alat penangkap citra misalnya kotoran debu yang menempel pada lensa foto maupun akibat proses pengolahan yang tidak sesuai. Komponen citra yang berfrekuensi rendah umumnya mempunyai nilai piksel konstan atau berubah sangat lambat (Munir, 2004). Contoh dari *noise* citra adalah *Gaussian noise* dan *Salt and Pepper noise*.

2.12 *Gaussian Noise*

Gaussian Noise dapat dibangkitkan dengan cara membangkitkan bilangan acak dengan distribusi *Gaussian*. Kemudian, untuk titik-titik yang terkena *noise*, nilai fungsi citra ditambahkan dengan nilai *noise* yang ada, atau dirumuskan di dalam persamaan (2.18) (Munir, 2004).

$$g(i, j) = f(i, j) + p \cdot a \quad (2.18)$$

dimana:

a adalah nilai bilangan acak berdistribusi *Gaussian* dan $a \sim N(0,1)$,

p adalah probabilitas *noise*,

$g(i, j)$ adalah nilai intensitas citra setelah terkena *noise*,

$f(i, j)$ adalah nilai intensitas citra sebelum terkena *noise*.

Adapun langkah-langkah dalam membangkitkan *gaussian noise* adalah sebagai berikut:

1. Menginput data citra, masing-masing *pixel* dinotasikan dengan $f(i, j)$. Data inputan merupakan citra RGB.
2. Mengubah parameter tiap-tiap *pixel* citra tersebut menjadi *grayscale* menggunakan persamaan (2.2).
3. Menentukan besar persen *noise*.
4. Mengubah seluruh intensitas citra menggunakan persamaan (2.18).

Untuk dapat membangkitkan bilangan acak berdistribusi *Gaussian*, tidak dapat langsung menggunakan fungsi *randomize*, tetapi memerlukan suatu metode untuk mengubah distribusi bilangan acak ke dalam fungsi f tertentu. Dalam aplikasi yang akan dirancang digunakan metode *rejection* untuk membangkitkan bilangan acak tersebut. Metode *rejection* dikembangkan dengan membangkitkan dua bilangan acak (x, y) dan ditolak bila $y > f(x)$ (Munir, 2004).

2.13 *Salt and Pepper Noise*

Salt & Pepper Noise dapat dibangkitkan dengan membangkitkan bilangan 255 (warna putih) pada titik-titik yang secara probabilitas lebih kecil dari nilai probabilitas *noise* (Munir, 2004).

$$g(i,j) \begin{cases} f(i,j) = 255, \text{ jika } \frac{p}{2} < b \leq p \\ f(i,j) = 0, \text{ jika } b \leq \frac{p}{2} \\ f(i,j) \text{ tidak berubah, jika } b > p \end{cases} \quad (2.19)$$

dimana:

$g(i,j)$ adalah nilai *grayscale* pada titik (i,j) ,

b adalah bilangan random berdistribusi Uniform dan $b \sim U(0,1)$,

p adalah probabilitas *noise*.

Adapun langkah-langkah dalam membangkitkan *salt and pepper noise* adalah sebagai berikut:

1. Menginput data citra, masing-masing *pixel* dinotasikan dengan $f(i,j)$. Data inputan merupakan citra RGB.
2. Mengubah parameter tiap-tiap *pixel* citra tersebut menjadi *grayscale* menggunakan persamaan (2.2).
3. Menentukan besar persen *noise*.
4. Membangkitkan bilangan random $b \sim U[0,1]$ yang memiliki distribusi uniform.
5. Mengubah seluruh intensitas citra menggunakan persamaan (2.19).

2.14 Fuzzy C-Means (FCM)

Fuzzy C-Means (FCM) klastering merupakan model pengelompokan *fuzzy* sehingga data dapat menjadi anggota dari semua kelas atau kluster terbentuk dengan derajat atau tingkat keanggotaan yang berbeda antara 0 hingga 1. Tingkat keberadaan dalam suatu kelas atau kluster ditentukan oleh derajat keanggotaannya. Konsep dasar FCM, pertama kali adalah menentukan pusat kluster yang akan menandai lokasi rata-rata untuk tiap-tiap kluster. Pada kondisi awal pusat kluster ini masih belum akurat. Tiap-tiap data memiliki derajat keanggotaan tiap-tiap data secara berulang, maka dapat dilihat bahwa pusat kluster akan menuju lokasi yang tepat. Perulangan ini didasarkan pada minimasi fungsi obyektif yang menggambarkan jarak dari titik data yang diberikan ke pusat dari kluster yang terboboti oleh derajat keanggotaan titik data dari himpunan *fuzzy* tersebut (Izakian dan Abraham, 2011).

Algoritma *Fuzzy C-Means* pertama kali diperkenalkan oleh Dunn (1974), kemudian dikembangkan oleh Bezdek (1981), kemudian direvisi oleh Rouben (1982), Trauwaert (1985), Gath dan Geva (1989), Gu dan Gubuisson (1990), Xie dan Benni (1991). Namun, algoritma FCM dari bezdek yang paling banyak digunakan. *Fuzzy C-Means* adalah salah satu teknik pengelompokan data yang diman keberadaan tiap titik data dalam suatu kelompok ditentukan oleh derajat keanggotaan, Berbeda degan *k-means clustering*, dimana suatu objek akan menjadi anggota satu klaster, dalam *Fuzzy C-Means* setiap objek dapat menjadi anggota beberapa klaster. Batas-batas dalam k-means adalah tegas (*hard*) sedangkan didalam *Fuzzy C-Means* adalah *soft*. *Fuzzy C-Means* bersifat sederhana, mudah diimplementasikan, memiliki kemampuan untuk mengelompokkan data yang besar, serta robust terhadap outlier (Izakian dan Abraham, 2011).

Konsep dasar FCM, pertama kali adalah menentukan pusat cluster yang akan menandai lokasi rata-rata untuk tiap-tiap cluster. Pada kondisi awal, pusat cluster ini masih belum akurat. Tiap-tiap data memiliki derajat keanggotaan untuk tiap-tiap cluster. Dengan cara memperbaiki pusat cluster dan nilai keanggotaan tiap-tiap data secara berulang, maka dapat dilihat bahwa pusat cluster akan menuju lokasi yang tepat. Perulangan ini didasarkan pada minimasi fungsi obyektif (Izakian dan Abraham, 2011). Tujuan dari FCM adalah meminimalkan fungsi objektif. Dimana fungsi objektif dirumuskan didalam persamaan (2.20):

$$J = \sum_{i=1}^N \sum_{j=1}^k u_{ij}^m ||\mathbf{x}_i - \mathbf{C}_j||^2 \quad (2.20)$$

dengan $1 \leq m \leq \infty$.

Berdasarkan persamaan (21) m adalah parameter *fuzzy* dan bernilai lebih dari 1, μ_{ij} adalah anggota dari kualifikasi *fuzzy* yang mengindikasikan anggota i sampai klaster ke j . X_i adalah data ke- i . C_j adalah pusat dari klaster. $||\mathbf{X}_i - \mathbf{C}_j||$ adalah *matrix* jarak pada data X_i terhadap setiap pusat klaster dengan mengambil jarak *Euclidean* antara titiknya terhadap pusat klaster. FCM merupakan algoritma yang baik untuk klastering, tetapi memiliki beberapa

kekurangan, yaitu waktu pengolahan yang lebih lama secara komputasional, selain itu sensitivitasnya terhadap inisialisasi nilai awal, dan sensitivitasnya terhadap *noise*. *Matrix* derajat keanggotaan \mathbf{U} dapat menghasilkan dari *fuzzy clustering* $\mathbf{U} = [u_{ij}]$, dimana:

$$\mathbf{U} = \begin{cases} \sum_{j=1}^k u_{ij} = 1, j = 1, 2, \dots, k \\ 0 < \sum_{i=1}^n u_{ij} < n, i = 1, 2, \dots, n \end{cases} \quad (2.21)$$

dengan fungsi objektif dari FCM adalah

$$J = \sum_{i=1}^N \sum_{j=1}^k u_{ij}^m ||\mathbf{x}_i - \mathbf{c}_j||^2 \quad (2.22)$$

dimana w_i adalah pusat kluster ke- i , $i=1,2,\dots, N$ dan $j=1,2,\dots,K; m \in (1, \infty)$ adalah indeks bobot. Fungsi objektif menunjukkan jumlah kuadrat dari bobot jarak dari tiap anggota data yang relevan terhadap pusat kluster. Untuk mendefinisikan derajat keanggotaannya sebagai berikut (Krishnan dan Ramamoorthy, 2014):

1. Menentukan pusat kluster. Dalam program yang akan dirancang pusat kluster ditentukan berdasarkan jumlah kluster yang kan digunakan.
2. Pusat awal klustering FCM $c_j(t)$, t =iterasi pertama. Untuk penentuan nilai dari pusat kluster tersebut di dalam program yang diinisialisasi menjadi $c_j(t) = \frac{j}{c} 255$.
3. Membentuk matriks \mathbf{U} dimana setiap elemennya akan di hitung derajat keanggotannya. Derajat keanggotaan dari u_{ij} , dengan $i=1,2,\dots,n$ dan $j=1,2,\dots,k$, dapat dihitung di dalam persamaan (2.23).

$$u_{ij}(k) = \frac{(d_{ij}(k))^{\frac{-1}{(m-1)}}}{\sum_{j=1}^k \left((d_{ij}(k))^{\frac{-1}{(m-1)}} \right)} \quad (2.23)$$

$d_{ij}(k)$ adalah jarak data x_i dengan pusat kluster C_j .

4. Merubah seluruh pusat kluster $c_i(t + 1)$, dalam persamaan (2.21).

$$c_j(t + 1) = \frac{\sum_{i=1}^n u_{ij}^m(k) x_i}{\sum_{i=1}^n u_{ij}^m(k)} \quad (2.24)$$

5. Menentukan optimal pusat kluster segment gambar seperti di dalam persamaan (2.25).

$$e = \sum_{i=1}^c ||c_i(t + 1) - c_i(t)||^2 \quad (2.25)$$

jika nilai $e < e_{max}$, algoritma berhenti. Sebaliknya, untuk $t=t+1$, kembali ke tahap 3 dan kembali melakukan iterasi. Berdasarkan nilai keanggotaan tertinggi yang didapatkan dilakukan *labeling* pada tiap-tiap *pixel* tujuannya adalah menentukan tiap-tiap *pixel* akan ditempati.

2.15 Expectation Maximization Algorithm

Expectation Maximization Algorithm merupakan Algoritma yang diperkenalkan oleh Dempster, Laird dan Rubin (1977). EM Algorithm merupakan sebuah metode optimisasi iteratif untuk estimasi Maksimum Likelihood (ML) yang berguna dalam permasalahan data yang tidak lengkap (*incomplete data*).

Dalam setiap iterasi pada EM Algorithm ini terdapat 2 tahap, yaitu tahap Ekspektasi atau tahap E (E-step) dan tahap Maksimisasi atau tahap M (M-step). Pada E-step dilakukan penghitungan nilai ekspektasi dari parameter data, dan M-step menghitung nilai estimasi parameter dengan menggunakan nilai ekspektasi yang ditemukan pada step sebelumnya. Kedua tahap tersebut akan terus dilakukan sampai mencapai nilai konvergen.

Ide dasar dari EM Algorithm adalah mengasosiasikan suatu *complete data problem* dengan *incomplete data problem* dengan tujuan agar secara komputasi menjadi lebih mudah. Tahapan tersebut ada pada E-step, dimana akan dibangun fungsi pengganti atau *surrogate function* yang melibatkan semacam

unobserved variabel yang akan diekspektasi. Dalam MM, surrogate function tersebut adalah Minorize yang akan dimaksimumkan pada tahap M-step. (Dempster, Laird dan Rubin, 1977)

E step bertujuan menemukan ekspektasi bersyarat dari *missing data* dengan syarat data yang diketahui nilainya (*observed*) dan penduga parameternya, kemudian mensubstitusikan nilai ekspektasi yang diperoleh terhadap *missing data*. Missing data disini merupakan fungsi dari obseravsi yang hilang (Y_{miss}) yg muncul dalam *complete data loglikelihood*, yaitu $\ell(\theta|Y)$.

$$Q(\theta|\theta^{(t)}) = E_{(Z|Y;\theta^{(t)})}[\ell(\theta; Y)] \quad (2.27)$$

$$= \int \ell(\theta; Y) f(Z|Y, \theta = \theta^{(t)}) dZ \quad (2.28)$$

kemudian, M step pada EM menentukan $\theta^{(t+1)}$ memaksimumkan ekspektasi loglikelihood tersebut.

$$Q(\theta^{(t+1)}|\theta^{(t)}) \geq Q(\theta|\theta^{(t)}), \text{ untuk semua } \theta$$

$$\theta^{(t+1)} = \arg \max_{\theta} Q(\theta|\theta^{(t)})$$

atau secara ringkas EM Algorithm diberikan sebagai berikut:

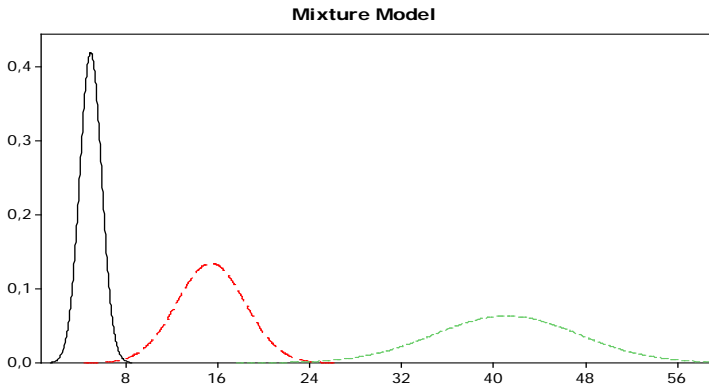
1. E-step : estimasi statistik cukup (*sufficient statistic*) untuk data lengkap Y_t dengan cara menghitung nilai ekspektasinya.
2. M-step: Tentukan $\theta^{(t+1)}$ dengan metode MLE (*Maximum Likelihood Estimation*) dari Y_t .
3. Iterasi sampai nilai $\theta^{(t)}$ konvergen, atau $|\theta^{(t+1)} - \theta^{(t)}| < \varepsilon$ dimana ε merupakan nilai yang sangat kecil mendekati 0.

2.16 *Expectation Maximization Algorithm* untuk Klastering

EM Algorithm dapat digunakan dalam berbagai ruang lingkup keilmuan statistika, karena EM Algorithm sangat mudah digunakan dengan proses iterasi yang hampir selalu mendekati konvergen. Dalam pengelompokkan atau klastering, pemilihan banyak cluster menjadi perhatian dari para peneliti. Metode K-means memberikan pemilihan banyak cluster yang bersifat

subyektif. Pendekatan lain yang dapat ditawarkan dalam klastering adalah metode admixture. Salah satunya yang paling sering digunakan adalah mixture model berbasis distribusi Normal atau yang dikenal dengan Gaussian Mixture Model (Dempster, Laird dan Rubin, 1977).

Mixture model secara jelas diilustrasikan pada Gambar 2.4. Jika setiap distribusi memiliki parameter tersendiri, dengan mean masing-masing μ_1, μ_2 , dan μ_3 maka dapat dikatakan bahwa model tersebut memiliki tiga cluster dengan centroidnya adalah μ_1, μ_2 , dan μ_3 .



Gambar 2.5 Densitas untuk mixture Normal

Dimisalkan pada Gambar, diketahui densitas untuk masing-masing distribusi memenuhi sifat:

$$f(x|\mu_1, \sigma_1^2) \sim N(\mu_1, \sigma_1^2),$$

$$f(x|\mu_2, \sigma_2^2) \sim N(\mu_2, \sigma_2^2),$$

$$f(x|\mu_3, \sigma_3^2) \sim N(\mu_3, \sigma_3^2).$$

Maka distribusi bersamanya adalah jumlahan dari masing-masing densitas Normal dengan pembobot atau mixed parameter $P_j, j = 1, 2, 3$. Jika diketahui pengamatan $x = x_1, x_2, x_3$ maka fungsi likelihoodnya adalah:

$$L(x_1, x_2, \dots, x_n | \mu_1, \dots, \mu_3, \sigma_1^2, \dots, \sigma_3^2, p_1, \dots, p_3) \\ = \prod_{k=1}^n \sum_{j=1}^3 p_j f(x_k | \mu_j, \sigma_j^2) \quad (2.29)$$

Jika secara umum terdapat sebanyak c kluster dan dituliskan parameter $\theta = \mu_1, \dots, \mu_c, \sigma_1^2, \dots, \sigma_c^2, p_1, \dots, p_c$, maka fungsi likelihood diberikan:

$$L(x_1, x_2, \dots, x_n | \theta) = \prod_{k=1}^n \sum_{j=1}^3 p_j f(x_k | \mu_j, \sigma_j^2) \quad (2.30)$$

dimana,

$$f(x_k | \mu_j, \sigma_j^2) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{1}{2}\left(\frac{x_k - \mu_j}{\sigma_j^2}\right)^2\right) \quad (2.31)$$

Persoalan dalam GMM adalah menentukan berapa peluang suatu titik pengamatan x_1 akan masuk ke cluster tertentu. Besaran peluang tersebut diukur melalui sebuah *hidden variable* Z . E-step dalam EM Algorithm adalah mengukur nilai ekspektasi dari *hidden variable* Z tersebut (Dempster, Laird dan Rubin, 1977).

Misal A adalah event atau kejadian dimana x_k berasal dari $f(x_k | \mu_j, \sigma_j^2) \sim N(x_k | \mu_j, \sigma_j^2)$. Dan D adalah datum dari x_k . Maka $E(Z)$ adalah $P(A|D)$, yang dapat dihitung melalui:

$$P(A|D) = \frac{P(A|D)P(A)}{P(D)}$$

Dengan $P(D)$ adalah distribusi peluang bersama *mixture model*. Sehingga dapat dituliskan menjadi:

$$E(Z) = (z_i | x_k, \theta_t) = \frac{f(x_k | z_i, \theta_t) f(z_i | \theta_t)}{f(x_k | \theta_t)} \\ = \frac{f(x_k | z_i, \mu_i(t) \sigma_i^2 \mathbf{I}) p_i(t)}{\sum_{j=1}^c f(x_k | z_j, \mu_j(t) \sigma_j^2 \mathbf{I}) p_j(t)} \quad (2.32)$$

Pada M-step dilakukan estimasi nilai parameter μ_j, σ_j^2, p_j . Fungsi diperoleh melalui MLE:

$$\mu_i(t+1) = \frac{\sum_k f(z_i | x_k, \theta_t) x_k}{\sum_k f(z_i | x_k, \theta_t)} \quad (2.33)$$

$$\sum_i (t+1) = \frac{\sum_k (z_i | x_k, \theta_t) [x_k - \mu_i(t+1)] [x_k - \mu_i(t+1)]^T}{\sum_k f(z_i | x_k, \theta_t)} \quad (2.34)$$

$$p_i(t+1) = \frac{\sum_k f(z_i | x_k, \theta_t)}{N} \quad (2.35)$$

langkah E-Step dan M-Step diulangi sampai konvergen.

2.17 Statistika Fitur Tekstur Citra

Untuk merepresentasikan konten citra medis, sebagian besar peneliti menggunakan fitur tekstur yang diekstrak secara global pada citra. Fitur tekstur dapat diekstrak secara struktural atau statistik. Secara struktural adalah tekstur suatu citra digambarkan dengan mengidentifikasi aturan penempatan dan secara statistik dengan menganalisa distribusi statistik dari intensitas citra. Analisa statistik dengan cara menganalisa distribusi dari intensitas citra tersebut (Haralick, 1973).

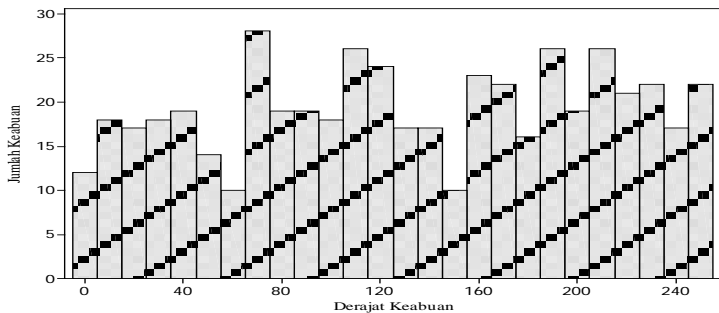
Hasil citra yang didapatkan merupakan sebuah *Gray Level Co-occurrence Matrices* (GLCM), hasil citra tersebut dibentuk menjadi sebuah *matrix* berukuran $N \times N$, tiap-tiap elemen *matrix* tersebut mengandung nilai intensitas abu-abu (*grayscale*). Dengan mendapatkan *matrix* tersebut dapat digunakan alat statistik untuk membandingkan masing-masing citra berdasarkan metode-metode yang ada disebut sebagai fitur tekstur citra, dimana fitur tekstur citra itu adalah nilai *energy*, *correlation*, *homogeneity*, dan *contrast* (Haralick, 1973).

2.18 Gray Level Co-occurrence Matrix (GLCM)

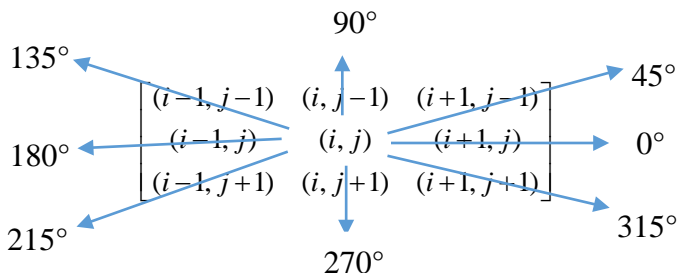
GLCM merupakan metode statistik dimana dalam perhitungan statistiknya menggunakan distribusi derajat keabuan dengan mengukur hubungan ketetanggaan antar *pixel* di dalam citra. Penggunaan statistik ini tidak terbatas, sehingga sesuai untuk tekstur-tekstur alami yang tidak terstruktur dari sub pola dan himpunan aturan (mikrostruktur). (Pradnyana, 2015) Metode statistik terdiri dari ekstraksi ciri orde pertama dan ekstraksi ciri orde kedua. Ekstraksi ciri orde pertama dilakukan melalui histogram citra sedangkan ekstraksi ciri statistik orde kedua

dilakukan dengan *matrix* kookurensi, yaitu suatu *matrix* yang merepresentasikan hubungan ketetanggaan antar *pixel* dalam citra pada berbagai arah orientasi dan jarak spasial. Ilustrasi ekstraksi ciri statistik ditunjukkan pada Gambar 2.5 dan Gambar 2.6.

Gambar 2.5 merupakan ilustrasi Histogram citra sebagai fungsi probabilitas kemunculan nilai intensitas pada citra, dan Gambar 2.6 merupakan ilustrasi hubungan ketetanggaan antar *pixel* sebagai fungsi orientasi dan jarak spasial pada citra (Pradnyana, 2015). Pada penelitian ini ciri statistik yang digunakan adalah ciri statistik orde dua. Salah satu teknik yang digunakan pada ciri ini adalah menghitung probabilitas hubungan ketetanggaan antara dua *pixel* pada jarak dan orientasi sudut tertentu.



Gambar 2.6 Contoh Histogram Derajat Keabuan dan Jumlah Keabuan



Gambar 2.7 Ilustrasi hubungan tetangga derajat antar *pixel*

Co-ocurrence berarti kejadian bersama, yaitu jumlah kejadian nilai *pixel* bertetangga dengan nilai *pixel* lain dalam jarak (d) dan orientasi sudut (θ) tertentu dimana jarak dinyatakan dalam *pixel* dan orientasi dinyatakan dalam derajat. Orientasi dibentuk dalam empat arah sudut dengan interval sudut 45° , yaitu 0° , 45° , 90° , dan 135° sedangkan jarak antar *pixel* biasanya ditetapkan sebesar 1 *pixel*, 2 *pixel*, 3 *pixel* dan seterusnya (Pradnyana, 2015).

Setiap titik (i,j) pada *matrix* kookurensi berorientasi berisi peluang kejadian *pixel* bernilai i bertetangga dengan *pixel* bernilai j pada jarak d serta orientasi dan $(180-\theta)$. Sebagai contoh pada persamaan (2.36) *matrix* A berukuran 5×5 memiliki *matrix* GLCM dengan ukuran yang sama. *matrix* kookurensi akan dihitung dengan nilai $d=1$ dan $\theta=0$ derajat. Jumlah frekuensi munculnya pasangan (i,j) dihitung untuk keseluruhan *matrix*. Jumlah kookurensi diisikan pada *matrix* GLCM pada posisi sel yang bersesuaian (Pradnyana, 2015).

$$\begin{bmatrix} 1 & 1 & 2 & 2 & 3 \\ 1 & 2 & 2 & 3 & 3 \\ 2 & 2 & 3 & 3 & 4 \\ 2 & 3 & 3 & 4 & 4 \\ 3 & 3 & 4 & 4 & 1 \end{bmatrix} \quad (2.36)$$

Dengan menggunakan contoh *matrix* A dan level pengelompokan warna dari 1 hingga 4 maka proses pembentukan *matrix* ketetanggaan dapat di jelaskan seperti pada persamaan (2.37).

$$\begin{bmatrix} 1 & 1 & 2 & 2 & 3 \\ 1 & 2 & 2 & 3 & 3 \\ 2 & 2 & 3 & 3 & 4 \\ 2 & 3 & 3 & 4 & 4 \\ 3 & 3 & 4 & 4 & 1 \end{bmatrix} \quad \begin{matrix} 1 & 2 & 3 & 4 \\ 1 & \begin{bmatrix} 2 & 2 & 0 & 1 \end{bmatrix} \\ 2 & \begin{bmatrix} 2 & 6 & 4 & 0 \end{bmatrix} \\ 3 & \begin{bmatrix} 0 & 4 & 8 & 3 \end{bmatrix} \\ 4 & \begin{bmatrix} 1 & 0 & 3 & 4 \end{bmatrix} \end{matrix} \quad (2.37)$$

Gambar 2.8 Pembentukan *Co-ocurrence Matrix*

Setelah *matrix* ketetanggaan terbentuk, *Matrix* tersebut akan dijadikan input untuk mencari *matrix* GLCM simetrisnya, proses ini dilakukan dengan menjumlahkan *matrix* tersebut dan *matrix* tranposenya (Pradnyana, 2015). Proses pembentukan *matrix* dijabarkan pada persamaan (2.38).

$$\begin{bmatrix} 2 & 2 & 0 & 1 \\ 2 & 6 & 4 & 0 \\ 0 & 4 & 8 & 3 \\ 1 & 0 & 3 & 4 \end{bmatrix} + \begin{bmatrix} 2 & 2 & 0 & 1 \\ 2 & 6 & 4 & 0 \\ 0 & 4 & 8 & 3 \\ 1 & 0 & 3 & 4 \end{bmatrix} = \begin{bmatrix} 4 & 4 & 0 & 2 \\ 4 & 12 & 8 & 0 \\ 0 & 8 & 16 & 6 \\ 2 & 0 & 6 & 8 \end{bmatrix} \quad (2.38)$$

Selanjutnya *matrix* simetris akan di normalisasi untuk dijadikan input dari parameter pengukuran probabilitas GLCM. Proses normalisasi *matrix* simetris dijabarkan didalam persamaan (2.39).

$$\begin{bmatrix} 4/80 & 4/80 & 0 & 2/80 \\ 4/80 & 12/80 & 8/80 & 0 \\ 0 & 8/80 & 16/80 & 6/80 \\ 2/80 & 0 & 6/80 & 8/80 \end{bmatrix} = \begin{bmatrix} 0.05 & 0.05 & 0 & 0.025 \\ 0.05 & 0.15 & 0.10 & 0 \\ 0 & 0.10 & 0.20 & 0.075 \\ 0.025 & 0 & 0.075 & 0.10 \end{bmatrix} \quad (2.39)$$

Untuk *matrix* GLCM citra *greyscale*, merupakan *matrix* berukuran 256 x 256, sesuai jumlah intensitas *grayscale* yang bernilai 0-255, dapat di ilustrasikan pada persamaan (2.40). Hasil normalisasi *matrix* tersebut, akan dijadikan input pengukuran probabilitas yang merepresentasikan ciri tekstur berdasarkan persamaan Haralick (Haralick, 1973). Persamaan tersebut adalah *contrast*, *correlation*, *energy*, *entropy* dan *homogeneity*.

$$p(a,b)= \begin{array}{ccccc} & 0 & 1 & 2 & 255 \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ \dots \\ 255 \end{array} & \left[\begin{array}{ccccc} p(0,0) & p(0,1) & p(0,2) & \dots & p(0,255) \\ p(1,0) & \dots & \dots & \dots & p(1,255) \\ p(2,0) & \dots & \dots & \dots & p(2,255) \\ \dots & \dots & \dots & \dots & \dots \\ p(255,0) & p(255,1) & p(255,2) & \dots & p(255,255) \end{array} \right] \end{array} \quad (2.40)$$

2.19 Energy (Angular Second Moment)

Parameter *energy* sering disebut dengan nilai keseragaman, dimana pengukuran nilai *energy* berdasarkan keseragaman dari tekstur yang ada, yaitu dari pasangan *pixel* yang berulang; gambar tersebut homogen atau secara tekstur gambar tersebut seragam *energy* dapat dirumuskan pada persamaan (2.41) (Haralick, 1973).

$$Energy = \sum_a \sum_b p_{ab}^2 \quad (2.41)$$

Dimana P_{ij} adalah probabilitas dari tiap-tiap intensitas pada ruang ke (i,j) . *Energy* adalah alat yang digunakan untuk mengukur tingkat *smooth* dari gambar. Jika distribusi dari P_{ij} makin seragam, maka nilai *energy* akan semakin kecil. Nilai *energy* akan semakin membesar bila pasangan pasangan pixel yang memenuhi syarat *matrix* intensitas kookurensi terkonsentrasi pada beberapa koordinat dan mengecil bila letaknya melebar.

2.20 Correlation

Correlation adalah pengukuran nilai depedensi linear pada citra, atau tingkat ketergantungan linear derajat keabuan dari *pixel-pixel* yang saling bertetangga pada citra. Nilai *correlation* dapat dirumuskan di dalam persamaan (2.46).

$$Correlation = \frac{\sum_{a=0}^{255} \sum_{b=0}^{255} a \cdot b \cdot P_{ab} - \mu_i \cdot \mu_j}{\sigma_i \cdot \sigma_j} \quad (2.42)$$

Dari persamaan μ_i dan μ_j adalah rata-rata dan σ_i dan σ_j adalah standar deviasi dari probabilitas marginal $P_a(i)$ dan $P_b(j)$ didapatkan dari menjumlah baris dan kolom dari *matrix* P_{ab} . Semakin tinggi nilai *correlation* (mendekati nilai 1)

mengimplikasikan hubungan yang linear antara tingkat keabuabuan (*grayscale*) dan pasangan *pixel*. (Haralick, 1973).

2.21 Homogeneity

Homogeneity adalah kehomogenan intensitas dalam citra, nilai *homogeneity* ini dikatakan dapat mempresentasikan kekasaran pada bidang citra. (kadir dan Susanto, 2013) nilai *Homogeneity* dapat dirumuskan di persamaan (2.43)

$$Homogeneity = \sum_{a=0}^{255} \sum_{b=0}^{255} \frac{1}{1 + (a - b)^2} P_{ab} \quad (2.43)$$

Dimana P_{ab} adalah probabilitas dari tiap-tiap intensitas pada ruang ke (a,b) . Nilai *Homogeneity* akan semakin membesar jika variasi intensitasnya dalam citra semakin mengecil (Haralick, 1973).

2.22 Contrast

Contrast merupakan nilai ukur untuk frekuensi spasial dari sebuah gambar dan perbedaan moment di GLCM. Perbedaannya merupakan perbedaan dari nilai terkecil dan nilai terbesar dari *pixel*. (Haralick, 1973) Persamaan *contrast* dapat dilihat dalam persamaan (2.44).

$$Contrast = \sum_a \sum_b (a - b)^2 \cdot P_{ab} \quad (2.44)$$

Dimana P_{ij} adalah probabilitas dari tiap-tiap intensitas pada ruang ke (i,j) . *Contrast* merepresentasikan variasi level *grayscale* dalam sebuah citra, biasanya *contrast* dijadikan sebagai parameter ketergantungan linear terhadap level *grayscale pixel* tetangga. *Contrast* juga bisa disebut jumlah kuadrat (*sum squares variance*) (Kadir dan Susanto, 2013).

2.23 Similarity

Performa dari suatu metode klastering adalah dengan menghitung nilai *similarity*-nya, *Similarity* adalah banyaknya *pixel*

yang benar saat dilakukan klastering kembali setelah citra diberikan *noise*. *Similarity* dapat dirumuskan pada persamaan (2.49) (Krishnan dan Ramamoorthy, 2014).

$$Similarity = \frac{\text{jumlah pixel yang benar}}{\text{total pixel}} \times 100\% \quad (2.45)$$

Jumlah *pixel* yang benar adalah jumlah *pixel* klastering dengan pemberian *noise* yang menempati kelas yang sama dengan kelas yang dihasilkan klastering tanpa pemberia *noise*.

2.24 Uji Normalitas Kolgomorov Smirnov

Uji kenormalan dilakukan sebelum menganalisis suatu permasalahan lebih lanjut. Jika data yang dianalisis berdistribusi bukan distribusi normal maka akan terjadi kesulitan dalam menurunkan distribusi sampling \bar{x} dan R.

Untuk menguji kenormalan suatu data dapat dilakukan dengan menggunakan uji Kolmogorov-Smirnov, karena data tersebut digunakan bersifat kontinyu.

Perumusan Hipotesis,

$H_0 : F_{0(x)} = F_{(x)}$ untuk semua nilai

$H_1 : F_{0(x)} \neq F_{(x)}$ untuk minimal satu x

$F_{0(x)}$ adalah fungsi distribusi yang dihipotesiskan dan $F_{(x)}$ merupakan fungsi distribusi yang belum diketahui.

Statistik uji : $D = \sum |S_{(x)} - F_{0(x)}| \quad (2.46)$

dimana $S_{(x)}$ adalah fungsi peluang kumulatif data.

Kesimpulan : Tolak H_0 jika $D > 1-\alpha$

Bila menggunakan paket program Minitab, maka uji kenormalan dapat ditinjau dari statistik uji *P-value*, dengan hipotesis

Perumusan Hipotesis,

$H_0 : \text{Data berdistribusi normal}$

$H_1 : \text{Data tidak berdistribusi normal}$

Kesimpulan : Tolak H_0 jika $Pvalue < 0,05$

2.25 *Silhouette Index*

Silhouette index (SI) merupakan ukuran validasi yang dapat memvalidasi sebuah data, klaster tunggal(satu klaster dari sejumlah klaster), atau bahkan keseluruhan klaster (Tan, Steinbach, dan Kumar, 2006). Untuk menghitung nilai SI dari sebuah data ke- i dibutuhkan 2 komponen yaitu a_i dan b_i . a_i adalah rata-rata jarak ke- i setiap semua data lainnya dalam satu klaster. Sedangkan b_i merupakan rata-rata jarak data ke- i terhadap data lainnya yang bukan termasuk dalam klasternya, kemudian diambil nilai terkecil. Nilai *Silhouette index* dirumuskan didalam persamaan (2.47) (Tan, Steinbach, dan Kumar, 2006).

$$SI_i^j = \frac{b_i^j - a_i^j}{\max(a_i^j, b_i^j)} \quad (2.47)$$

Dengan nilai a_i^j dan b_i^j dirumuskan didalam persamaan (2.48) dan persamaan (2.49).

$$a_i^j = \frac{1}{m_n - 1} \sum_{\substack{r=1 \\ r \neq i}}^{m_n} d(x_i^j, x_r^i), i=1,2,..., m_j \quad (2.48)$$

$$b_i^j = \min_{\substack{n=1,...,k \\ n \neq j}} \left\{ \frac{1}{m_n} \sum_{\substack{r=1 \\ r \neq i}}^{m_n} d(x_i^j, x_r^n) \right\}, i=1,2,..., m_n \quad (2.49)$$

Dimana:

$d(x_i^j, x_r^i)$ = jarak data ke- i dengan data ke- r

m_j = jumlah data dalam klaster ke- j

Nilai a_i^j mengukur seberapa tidak miripnya sebuah data dengan klaster yang diikutinya. Nilai b_i^j mengukur seberapa tidak miripnya sebuah data dengan klaster lainnya. Untuk nilai SI sebuah klaster adalah nilai rata-rata SI semua data yang bergabung dalam klaster tersebut, dapat dirumuskan didalam persamaan (2.50) dan untuk nilai SI global adalah rata-rata semua SI klaster dapat dirumuskan didalam persamaan (2.51) (Tan, Steinbach, dan Kumar, 2006).

$$SI_j = \frac{1}{m_j} \sum_{i=1}^{m_j} SI_i^j \quad (2.50)$$

$$SI = \frac{1}{k} \sum_{j=1}^k SI_j \quad (2.51)$$

Nilai SI inilah yang menjadi validitas dari klaster. Nilai SI dan mempunyai range $[-1,1]$. Nilai yang semakin mendekati 1 menandakan semakin tepatnya data tersebut berada dalam klaster yang ditempatinya. Nilai SI bernilai negatif menandakan data tersebut tidak tepat didalam klaster yang ia tempati. Nilai SI bernilai=0 menandakan data tersebut berada di perbatasan antara dua klaster.

2.26 *Partition Coefficient Index*

Dalam validasi klastering *fuzzy* dapat dilakukan menggunakan dengan menghitung nilai *Partition Coefficient Index* (PCI). Nilai ini didapat dengan mengikuti konsep *fuzzy* yaitu semua data dapat menjadi anggota semua klaster dengan nilai derajat keanggotaan yang dimilikinya. Bezdek (1981) mengusulkan validitas dengan menghitung nilai PCI sebagai evaluasi nilai derajat keanggotaan, tanpa memandang nilai vektor (data) yang biasanya mengandung informasi geometrik (sebaran data). Nilai PCI dapat dirumuskan didalam persamaan (2.52) (Bezdek, 1981).

$$PCI = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^K u_{ij}^2 \quad (2.52)$$

Dimana:

u_{ij}^2 = nilai derajat keanggotaan satu *pixel*

Nilai PCI berada dalam rentang $[0,1]$, nilai semakin mendekati 1 mempunyai arti bahwa kualitas klaster yang didapat semakin baik.

BAB III

METODOLOGI PENELITIAN

3.1 Sumber Data

Penelitian ini akan menggunakan data sekunder, data tersebut merupakan citra MRI data rekam medis di Rumah Sakit Umum Daerah Soetomo, Surabaya. Citra MRI tersebut merupakan data rekam medis MRI yang dipilih berdasarkan rekomendasi dari sudut pandang medis. Data yang dipakai adalah *sequence ax T2 Flair slice* ke 18 dan *ax T1 memp+C slice* ke 18.

3.2 Ruang Implementasi

Lingkungan implementasi program terdiri atas lingkungan perangkat keras dan lingkungan perangkat lunak. Lingkungan perangkat keras dan lingkungan perangkat lunak yang digunakan untuk membangun sistem dijabarkan pada sub bab 3.2.1 dan 3.2.2.

3.2.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam membangun sistem ini memiliki spesifikasi seperti yang ditunjukkan oleh Tabel 3.1.

Tabel 3.1 Spesifikasi Perangkat Keras

Perangkat keras	Prosesor	Intel (R) Core (TM) i5-3230 CPU @ 2.60GHz(4CPUs),~2.6 GHz
	Memori RAM	8.00 GB

3.2.2 Lingkungan Perangkat Lunak

Beberapa perangkat lunak yang digunakan dalam membangun sistem ini adalah sebagai berikut:

1. Sistem Operasi Windows 8.1 Enterprise 64-bit.
2. Matlab 9.0 R2016a.

3.3 Variable Penelitian dan Struktur Data

Pada bagian ini akan menjelaskan Variable Penelitian dan Struktur Data dari rekam medis MRI yang digunakan. Variabel yang digunakan ditampilkan pada Tabel 3.1.

Tabel 3.2 Variabel Penelitian

Variabel	Keterangan	Skala data
<i>Grayscale</i>	Variabel yang digunakan merupakan besarnya intensitas keabuan di tiap-tiap <i>pixel</i> citra MRI	Interval

Variabel *Grayscale* mewakili intensitas keabuan pada tiap-tiap *pixel* citra MRI, dimana intensitas citra MRI dapat direpresentasikan menjadi sebuah *matrix* berukuran $M \times N$ dijelaskan pada persamaan (1). Elemen *Matrix* tersebut merupakan besar intensitas keabuan yang diwakili dengan nilai 0-255, sehingga tiap-tiap besaran element akan dikumpulkan dan dipisahkan berdasarkan besar intensitasnya. Lalu dibentuk menjadi sebuah struktur data, yang dapat dilihat berdasarkan Tabel 3.2.

Tabel 3.3 Struktur Data

Kolom i	Kolom j	Intensitas
1	1	$x(i,j)$
1	2	$x(i,j)$
...
1	M	$x(i,j)$
...
...
M	N	$x(i,j)$

Berdasarkan struktur data ini, akan dilakukan analisis klastering yang melibatkan tingkat intensitas *grayscale* dan frekuensinya didalam suatu citra.

3.4 Langkah Analisis

Tahapan yang dilakukan dalam penelitian ini setelah didapat data citra rekam medis MRI adalah sebagai berikut.

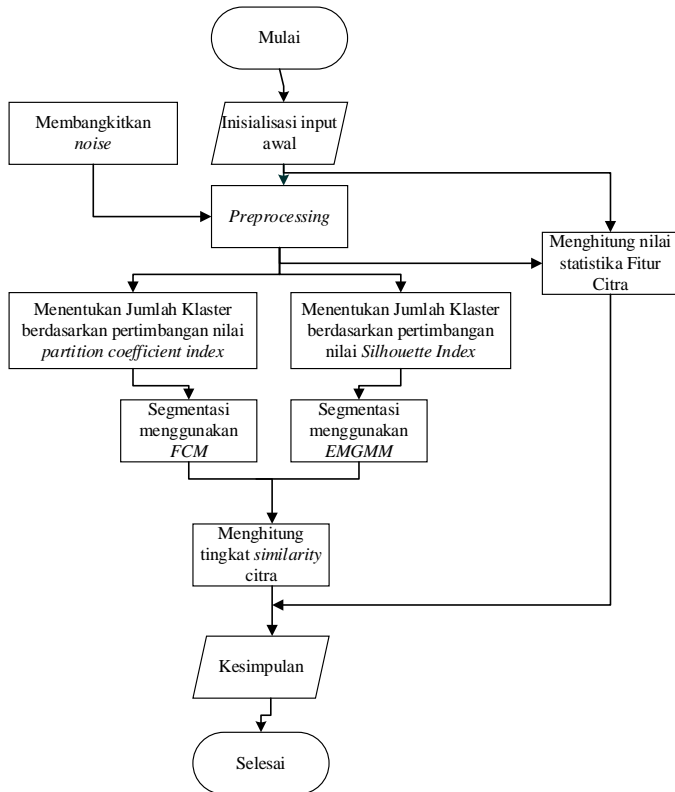
1. Input data citra MRI.
2. *Preprocessing* data citra MRI dengan tahapannya sebagai berikut.

- a) Melakukan proses *thresholding*, di dalam proses ini *pixel* citra yang nilainya kurang dari atau sama dengan nilai *threshold* akan diganti dengan warna hitam. Sedangkan, *pixel* citra yang nilainya lebih dari *threshold* akan diganti dengan putih. Data keluaran dari proses *thresholding* ini berupa citra biner.
 - b) Menentukan *head contour*, penentuan *head contour* dilakukan dengan cara mencari *pixel* yang berwarna putih pada setiap baris. *Pixel* yang berwarna putih pertama adalah *head contour*.
 - c) Melakukan proses *head masking* dengan cara *blocking area head contour*.
 - d) Melakukan substraksi antara *head mask* dan *threshold*. Citra *head mask* dikurangi dengan citra *threshold*.
 - e) Melakukan proses *morphological opening* untuk menghilangkan *pixel* yang kecil.
 - f) Melakukan proses deteksi tepi menggunakan *operator sobel* agar *pixel-pixel* area otak tidak terputus.
 - g) Menggabungkan citra hasil deteksi tepi dan citra setelah proses *morphological opening*. Proses ini bertujuan untuk menggabungkan titik-titik yang terputus.
 - h) Melakukan *blocking area* kepala setelah proses penggabungan.
 - i) Menentukan area otak dengan cara substraksi.
 - j) Mengembalikan *pixel-pixel* yang bernilai 1 dengan nilai intensitasnya.
3. Menghitung nilai Statistika Fitur Tekstur Citra (*energy*, *correlation*, *homogeneity*, dan *contrast*).
 4. Menentukan jumlah klaster dengan melihat nilai *silhouette index* untuk EM-GMM dan nilai *partition coefficient index* untuk FCM
 5. Segmentasi citra MRI menggunakan FCM.
 6. Segmentasi citra MRI menggunakan EM-GMM.

7. Membangkitkan *noise*, dengan metode *gaussian noise* dan *pepper and salt noise* dengan tingkat 1%, 5%, 10% pada citra MRI.
8. Segmentasi citra MRI yang telah diberi *gaussian noise* dengan tingkat 1%, 5%, 10% menggunakan FCM.
9. Segmentasi citra MRI yang telah diberi *gaussian noise* dengan tingkat 1%, 5%, 10% menggunakan EM-GMM.
10. Menghitung dan membandingkan nilai *similarity* masing-masing metode segmentasi pada tiap-tiap tingkat *noise*.
11. Membuat kesimpulan.

3.5 Diagram Alir

Berdasarkan langkah penelitian, dapat dibangun Diagram Alir penelitian secara umum terdapat di dalam Gambar 3.1.



Gambar 3.1 Flowchart langkah-langkah yang akan dilakukan dalam penelitian

BAB IV

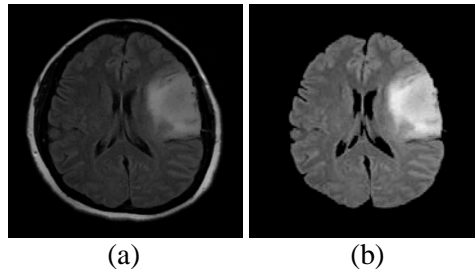
IMPLEMENTASI, UJI COBA, ANALISIS DAN PEMBAHASAN

4.1 Pemilihan Data dan *Preprocessing*

Pada Tahap ini dilakukan *preprocessing* data rekam medis MRI. Tujuan *preprocessing* adalah untuk menghilangkan bagian tengkorak pada citra MRI agar tidak mengganggu proses segmentasi. Setelah dilakukan *preprocessing* akan dilakukan perhitungan nilai statistika fitur citra sebelum dan sesudah *preprocessing*. Pemilihan citra dilakukan berdasarkan rekomendasi dari sudut pandang medis. Yaitu, berdasarkan penanganan pihak dokter dengan pemberian *contrast media* pada pasien. Pemberian *contrast media* akan menghasilkan citra MRI yang lebih fokus pada tumor otak, sehingga pembengkakan sekitar area tumor dapat diminimalisasi di penampakkannya. Berdasarkan rekomendasi tersebut maka dilakukan analisa pada *sequence ax T2 flair* dan *sequence ax T1 memp+c*. Data citra MRI yang didapatkan adalah berupa sebuah matriks yang mempunyai skala data interval atau bilangan cacah dan terbaca *uint8* didalam program dan mempunyai *range* sebesar 0-255. Selanjutnya akan dilakukan *preprocessing* data rekam medis MRI.

4.1.1 Data Masukan *Preprocessing*

Contoh data masukan pada *preprocessing* adalah citra *grayscale 2D sequence ax T2 Flair slice* ke 18 setentang dengan *sinus maxillaris* atau sejajar dengan ujung hidung, citra tersebut memiliki ekstensi *.dcm*. Dalam Gambar 4.1 (a), terlihat bahwa citra masukan masih berupa area otak dengan tulang kepala. Data keluaran dari *preprocessing* ini adalah citra *grayscale 2D* yang hanya terlihat area otak saja tanpa tulang kepala. Contoh data keluaran dari *preprocessing* dapat dilihat dalam Gambar 4.1 (b). Data keluaran dari *preprocessing* akan menjadi data masukan dari proses inti, yaitu segmentasi. Tujuan penghilangan image tulang kepala agar proses segmentasi hanya fokus pada area otak.



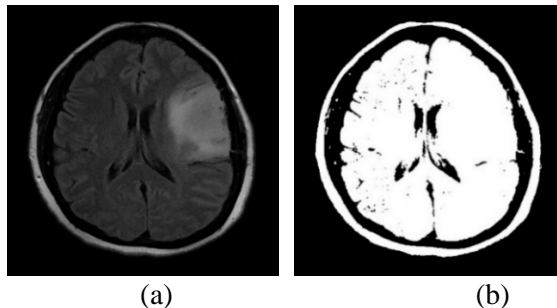
Gambar 4.1 Data Citra masukan dan keluaran *preprocessing sequence ax T2 Flair*

Langkah-langkah dalam proses *preprocessing* ini dapat dijabarkan di dalam Algoritma 4.1.

Algoritma 4.1 *Preprocessing*

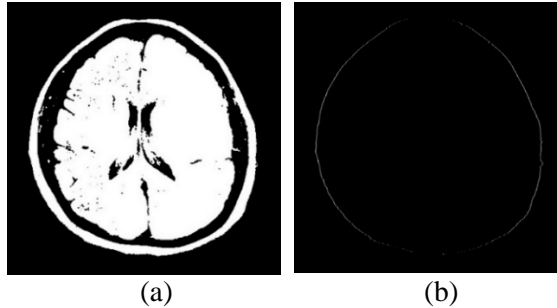
1. Lakukan proses *Thresholding*
 Proses *Thresholding* dilakukan dengan menggunakan metode Ridler-Calvard. *Pixel* citra yang nilainya kurang dari satu atau sama dengan nilai *threshold* akan diganti dengan warna hitam. Sedangkan *pixel* citra yang lebih besar dari nilai *threshold* akan diganti dengan warna putih. Data keluaran proses *thresholding* ini berupa citra biner. Hasil dari proses ini dapat dilihat pada Gambar 4.3.
2. Tentukan *head contour*
 Penentuan *head contour* dilakukan dengan cara mencari *pixel* yang berwarna putih pada setiap baris. *Pixel* yang berwarna putih pertama adalah *head contour*. Hasil *head contour* dapat dilihat pada Gambar 4.4
3. Lakukan proses *head masking* dengan cara *blocking area head contour*. Hasil *head masking* dapat dilihat pada Gambar 4.5
4. Melakukan substraksi antara *head mask* dan *threshold*. Citra *head mask* dikurangi dengan citra *threshold*. Hasil Substraksi dapat dilihat pada Gambar 4.6.
5. Melakukan proses *morphological opening* untuk menghilangkan *pixel* yang kecil. Hasil proses *morphological opening* dapat dilihat pada Gambar 4.7

6. Melakukan proses deteksi tepi menggunakan *operator sobel* agar *pixel-pixel* area otak tidak terputus. Hasil proses deteksi tepi dapat dilihat pada Gambar 4.8.
 7. Menggabungkan citra hasil deteksi tepi dan citra setelah proses *morphological opening*. Proses ini bertujuan untuk menggabungkan titik-titik yang terputus. Hasil proses penggabungan citra hasil deteksi tepi dan citra setelah proses *morphological opening* dapat dilihat pada Gambar 4.9.
 8. Melakukan *blocking* area kepala setelah proses penggabungan. Hasil proses *blocking* area kepala dapat dilihat pada Gambar 4.10.
 9. Menentukan area otak dengan cara substraksi. Hasil proses penentuan area otak dapat dilihat pada Gambar 4.11.
 10. Mengembalikan *pixel-pixel* yang bernilai 1 dengan nilai intensitasnya. Hasil output dapat dilihat pada Gambar 4.12.
- Implementasi Algoritma 4.1 ke dalam data MRI dihasilkan berturut-turut didalam Gambar 4.2 sampai dengan Gambar 4.11



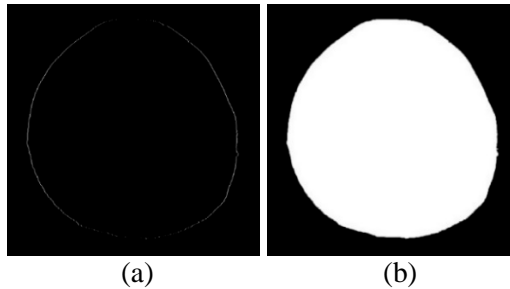
Gambar 4.2 Citra Hasil setelah di *Thresholding* (a) Citra Awal (b) Citra Setelah di-*threshold*

Gambar 4.2 bagian a menunjukkan citra MRI masukan sebelum dilakukan proses *thresholding*, sedangkan citra hasil *thresholding* ditunjukkan pada Gambar 4.2 bagian b. *Pixel* citra dengan intensitas keabuan kurang dari atau sama dengan nilai *threshold* akan di ganti dengan warna hitam, sedangkan *pixel* citra yang intensitasnya lebih dari nilai *threshold* akan diganti dengan warna putih.



Gambar 4.3 Citra Hasil *Head Contour* (a) Citra Asal *Thresholding* (b) Citra Setelah di tentukan *Head Contour*

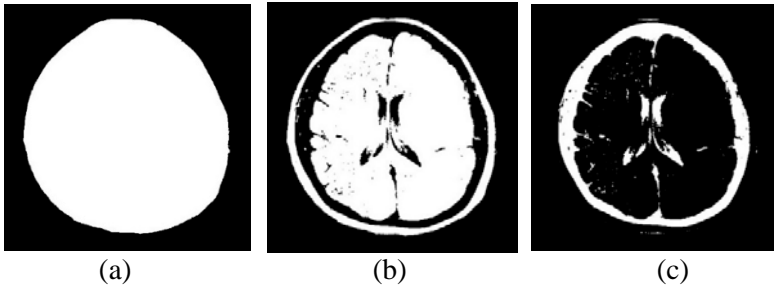
Gambar 4.3 bagian b menunjukkan citra setelah ditentukan *head contour*-nya. *Head Contour* merupakan batas antara citra dengan *background*. *Head Contour* ditentukan dengan cara mencari *pixel* pertama dan terakhir yang berwarna putih pada setiap barisnya. Selain *pixel* putih pertama dan terakhir akan dibiarkan berwarna hitam. Penentuan *contour* kepala dengan cara ini lebih efektif dibandingkan dengan deteksi tepi, karna yang dibutuhkan hanya batas-batas luar kepala saja. Citra awal sebelum ditentukan *Head Contour*-nya ditunjukkan oleh gambar 4.3 bagian a.



Gambar 4.4 Citra Hasil *Head Masking* (a) Citra Asal *Head Contour* (b) Citra Setelah di *blocking* dan ditentukan *Head Masking*-nya

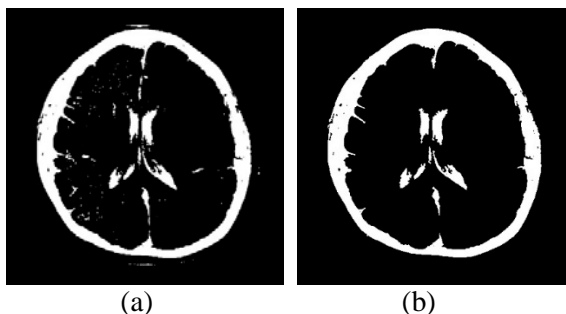
Gambar 4.4 bagian b menunjukan hasil proses *head masking*. *Head masking* dilakukan dengan cara melakukan proses *blocking* pada citra *head contour* yang ditunjukkan oleh Gambar

4.4 bagian a. *Pixel* yang berwarna putih pada *head masking* merupakan area kepala, sedangkan *pixel* yang berwarna hitam merupakan area *background*.



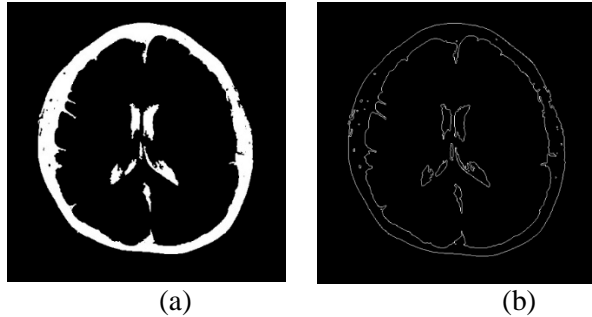
Gambar 4.5 Citra Hasil *Substraksi* dengan pengurangan *Head Mask* dengan citra *Threshold* (a) Citra *Head Mask* (b) Citra *Threshold* (c) Citra hasil *Substraksi*

Gambar 4.5 bagian c menunjukkan hasil proses substraksi antara *head mask* dan citra input yang telah di-*threshold*. Citra *head mask* dan citra input yang telah di tunjukkan oleh Gambar 4.5 bagian a, sedangkan hasil *threshold* ditunjukkan oleh Gambar 4.5 bagian b. Pada Gambar 4.6 bagian c menunjukkan area otak dan *background* berwarna hitam, sedangkan area tulang kepala berwarna putih.



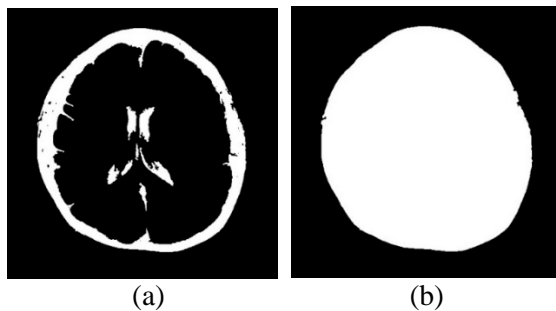
Gambar 4.6 Citra Hasil *Opening* (a) Citra Asal *Substraksi* (b) Citra Setelah di-*opening*

Gambar 4.6 merupakan hasil morfologi *opening* citra hasil substraksi. Gambar 4.6 bagian a adalah citra sebelum di-*opening* dan Gambar 4.6 bagian b adalah citra setelah di-*opening*. Tujuannya adalah menghilangkan *pixel-pixel* yang kecil.



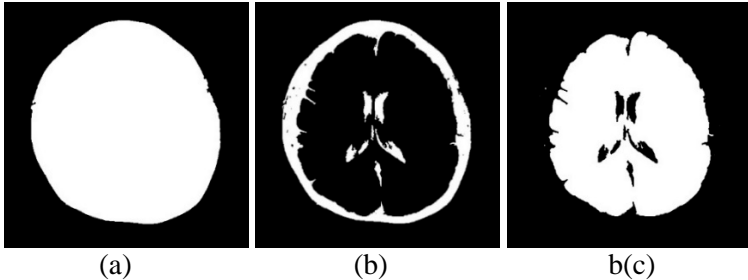
Gambar 4.7 Citra Hasil Proses Deteksi Tepi (a) Citra Asal *Opening* (b) Citra Setelah Proses Deteksi Tepi

Gambar 4.7 bagian b merupakan citra hasil proses deteksi tepi, sedangkan Gambar 4.7 bagian a adalah merupakan citra sebelum dideteksi tepinya. Tujuan dari tahap ini adalah untuk menyambung *pixel-pixel* area otak yang terputus, sehingga penentuan area otak lebih akurat.



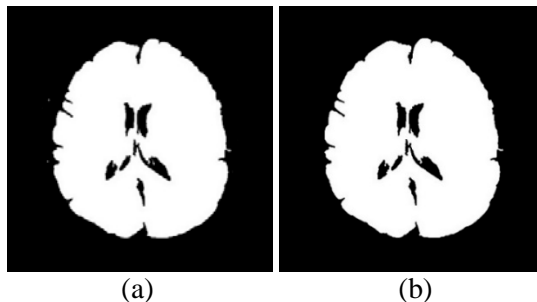
Gambar 4.8 Citra Hasil Proses *Blocking* (a) Citra Asal *Opening* (b) Citra Setelah Proses *Blocking*

Gambar 4.8 bagian a merupakan citra asal sebelum dilakukan proses *blocking*. Gambar 4.8 bagian b merupakan citra setelah dilakukan proses *blocking*. Citra hasil *blocking* akan digunakan untuk proses subtraksi penentuan area otak.



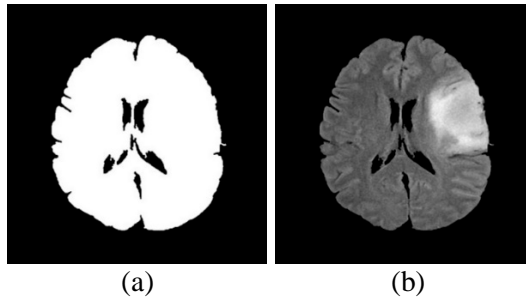
Gambar 4.9 Citra Hasil *Substraksi* dengan pengurangan *blocking* dengan citra *opening* (a) Citra *Head Mask* (b) Citra *Threshold* (c) Citra hasil *Substraksi*

Gambar 4.9 bagian a merupakan citra hasil *blocking*. Citra hasil *blocking* tersebut akan dikurangkan dengan citra pada Gambar 4.9 bagian b. Hasil dari proses pengurangan ini adalah area otak yang masih terdapat *pixel* kecil seperti yang ditunjukkan pada Gambar 4.9 bagian c.



Gambar 4.10 Citra Hasil Proses *Opening* (a) Citra Asal *Substraksi* (b) Citra Setelah Proses *Opening*/Citra Biner Hasil *Preprocessing*

Gambar 4.10 bagian a merupakan citra hasil substraksi yang belum dihilangkan *pixel* yang kecil sehingga perlu dilakukan *opening*. *Pixel* kecil ini merupakan *noise* sehingga perlu dihilangkan dengan cara di *opening*. Gambar 4.10 bagian b merupakan citra yang sudah di *opening*, dan citra ini sudah tidak mengandung *noise*.

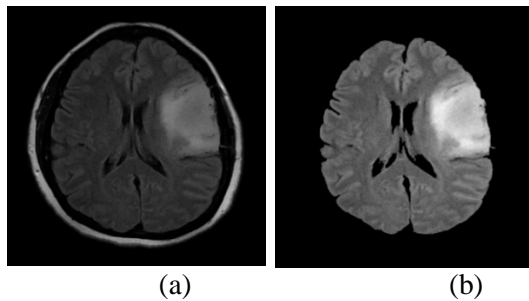


Gambar 4.11 Citra Hasil *preprocessing* (a) Citra Biner Hasil *Preprocessing* (b) Citra *grayscale* Hasil *Preprocessing*

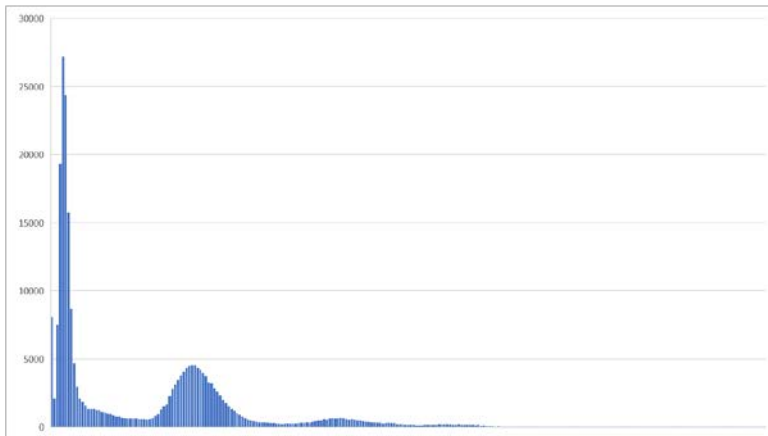
Gambar 4.11 bagian a merupakan citra MRI hasil *preprocessing* yang sudah tidak ada bagian tulangnya, sehingga citra ini akan dikembalikan *pixel* putihnya menjadi *pixel* sesuai *grayscale* data awalnya. Gambar 4.11 bagian b merupakan citra *grayscale* keluaran *preprocessing* yang sudah siap untuk menjadi data masukan untuk proses segmentasi. Data yang dihasilkan di dalam *preprocessing* adalah berupa sebuah matriks yang memiliki skala data ordinal atau bilangan cacah dan terbaca *uint8* di dalam program. Data ini mempunyai *range* sebesar 0-255. Saat memasuki proses segmentasi, data tersebut akan dirubah menjadi bilangan real positif dan terbaca *double* di dalam program. Data ini mempunyai *range* sebesar 0-255. Tujuan merubah jenis bilangan pada data tersebut adalah agar data tersebut dapat diolah menggunakan FCM dan EMGMM.

4.1.2 Histogram dan Statistika Fitur Tekstur Citra

A. Pada subbab ini akan diterangkan Histogram citra MRI. Data pertama yang akan di analisis adalah *sequence ax T2 flair*. Penentuan histogram dan nilai statistika fitur citra berdasarkan citra sebelum di-*preprocessing* dan citra hasil dari *preprocessing*. Citra sebelum dan sesudah tahapan *preprocessing* ditampilkan pada Gambar 4.12 (a) dan (b). Histogram citra sebelum dilakukan *preprocessing* ditampilkan pada Gambar 4.13.



Gambar 4.12 Citra Hasil *preprocessing* (a) Citra sebelum *preprocessing*
(b) Citra *grayscale* Hasil *preprocessing*



Gambar 4.13 Histogram Citra Sebelum *preprocessing*

Gambar 4.12 menunjukkan histogram citra sebelum dilakukan *preprocessing* memiliki 4 puncak, yang dicurigai pada puncak dengan intensitas paling tinggi tersebut akan membuat sebuah *cluster* baru, sehingga harus dihapuskan dengan tahapan *preprocessing*. Setelah dilakukan *preprocessing* data citra tersebut didapatkan histogram yang ditampilkan pada Gambar 4.13.



Gambar 4.14 Histogram Citra Setelah *preprocessing*

Data inilah yang akan dilanjutkan dalam tahap segmentasi. Selain melakukan visualisasi citra menggunakan histogram, akan dihitung nilai statistika fitur tekstur citra sebelum dan setelah proses *preprocessing*. Nilai statistika fitur tekstur citra sebelum *preprocessing* ditampilkan pada Tabel 4.1.

Tabel 4.1 Nilai Statistika Fitur Tekstur Citra Sebelum *preprocessing sequence ax T2 flair*, a) Jarak 1 Pixel, b) Jarak 2 Pixel, c) Jarak 3 Pixel

a) Jarak 1 pixel					
Sudut	0	45	90	135	rataan
<i>contrast</i>	0.061583	0.089407	0.056316	0.086657	0.073491
<i>correlation</i>	0.96884	0.9548	0.9715	0.95619	0.96283
<i>energy</i>	0.38099	0.37339	0.38248	0.37403	0.37772
<i>homogeneity</i>	0.97157	0.9622	0.97322	0.96286	0.96746

b) Jarak 2 <i>pixel</i>					
Sudut	0	45	90	135	rataan
<i>contrast</i>	0.14519	0.21148	0.12312	0.20351	0.17083
<i>correlation</i>	0.9266	0.89329	0.93775	0.89731	0.91374
<i>energy</i>	0.36416	0.35404	0.36657	0.35534	0.36003
<i>homogeneity</i>	0.95002	0.93843	0.95274	0.93945	0.94516
c) Jarak 3 <i>pixel</i>					
Sudut	0	45	90	135	rataan
<i>Contrast</i>	0.245	0.34238	0.19671	0.3258	0.27747
<i>correlation</i>	0.87626	0.82758	0.90065	0.83593	0.86011
<i>Energy</i>	0.35259	0.34087	0.35538	0.34269	0.34788
<i>homogeneity</i>	0.93539	0.92358	0.93898	0.92567	0.9309

Dari Tabel 4.1 didapatkan nilai statistika fitur tekstur citra berdasarkan perhitungan dari GLCMnya. Pada jarak *pixel* 1 dan pada arah sudut 0, 45, 90, dan 135 derajat, didapatkan nilai rataan *contrast* sebesar 0,073491 yang berarti pada jarak 1 *pixel* tersebut memiliki ukuran variasi antar derajat keabuan yang rendah atau dapat dikatakan juga berdasarkan nilai tersebut letak intensitas *pixel-pixel*nya tidak terlalu jauh dari diagonal utama sehingga ukuran variasinya rendah. Pada jarak 2 *pixel* didapatkan nilai rataan *contrast* sebesar 0,17083, nilai tersebut menunjukkan ukuran variasi yang lebih besar terhadap diagonal utamanya dibandingkan jarak 1 *pixel*. Pada jarak 3 *pixel* didapatkan nilai rataan *contrast* sebesar 0,27747, terdapat peningkatan nilai *contrast* dibandingkan pada jarak 1 dan 2 *pixel*, sehingga terdapat peningkatan ukuran variasi pada diagonal utama untuk jarak 3 *pixel*.

Untuk nilai *correlation* pada jarak 1 *pixel* didapatkan nilai rataan sebesar 0,96283, hal ini menunjukkan pada jarak 1 *pixel* ukuran ketergantungan linear derajat keabuan citra cukup tinggi karena mendekati nilai 1. Pada jarak 2 *pixel* didapatkan nilai

correlation sebesar 0,91374 terjadi penurunan nilai rata-ran *correlation* dapat dikatakan untuk jarak 2 *pixel* terjadi penurunan ketergantungan linear derajat keabuan citra, tetapi nilai yang didapatkan masih cukup tinggi sebab mendekati nilai 1, sehingga ketergantungan linear derajat keabuan tiap *pixel*-nya masih tinggi. Untuk *correlation* pada jarak 3 *pixel*, didapatkan nilai sebesar 0,86011 hal ini menunjukkan penurunan nilai ketergantungan linear antar *pixel-pixel*nya, tetapi nilainya masih cukup tinggi sebab mendekati nilai 1, sehingga masih memberikan petunjuk struktur linear dalam citra tersebut.

Nilai rata-ran *Energy* pada jarak 1 *pixel* didapatkan sebesar 0,37772, dapat dikatakan untuk jarak 1 *pixel* citra tersebut memiliki keteraturan yang cukup rendah, sebab nilainya cenderung mendekati nilai 0 atau dapat dikatakan *pixel-pixel* pada citra tersebut strukturnya tidak teratur atau bervariasi. Pada jarak 2 *pixel* didapatkan rata-ran nilai *energy* sebesar 0,36003 terjadi penurunan pada jarak 1 *pixel*, tetapi tidak terlalu jauh besarnya, dapat dikatakan pada jarak 2 *pixel*, citra tersebut juga memberikan struktur yang tidak teratur tetapi juga menurunkan nilai variasi *pixel*-nya. Pada jarak 3 *pixel* didapatkan rata-ran nilai *energy* sebesar 0,34788, dimana terjadi penurunan tingkat *energy*, nilai tersebut memberikan kesimpulan pada jarak 3 *pixel*, struktur citra tersebut memiliki struktur yang tidak teratur tetapi mengalami penurunan variasi dibandingkan pada jarak 1 *pixel* dan 2 *pixel*.

Nilai rata-ran *homogeneity* untuk jarak 1 *pixel* didapatkan nilai sebesar 0,96746 yang berarti untuk jarak 1 *pixel* terdapat tingkat homogen yang tinggi sebab mendekati nilai 1, atau dapat disimpulkan juga pada jarak 1 *pixel* letak intensitasnya jauh terhadap diagonal utamanya. Pada jarak 2 *pixel* nilai rata-ran *homogeneity*-nya didapatkan sebesar 0,94516 nilai tersebut mengalami penurunan dari jarak 1 *pixel*, tetapi nilai ini masih memberikan tingkat homogen yang tinggi antar *pixel*-nya. Pada jarak 3 *pixel* didapatkan nilai *homogeneity* sebesar 0,9309 nilai tersebut masih menunjukkan tingkat homogen yang tinggi antar *pixel-pixel* didalam citra tersebut. Setelah melakukan analisa statistika fitur

tekstur citra sebelum *preprocessing*, akan dilakukan juga perhitungan terhadap statistika fitur tekstur citra setelah *preprocessing* yang ditampilkan pada tabel 4.2.

Tabel 4.2 Nilai Statistika Fitur Tekstur Citra Setelah *preprocessing* a) Jarak 1 Pixel, b) Jarak 2 Pixel, c) Jarak 3 Pixel

a) Jarak 1 <i>pixel</i>					
Sudut	0	45	90	135	rataan
<i>contrast</i>	0.10103	0.14026	0.090815	0.13451	0.11665
<i>correlation</i>	0.98198	0.97501	0.98385	0.97603	0.9792
<i>energy</i>	0.4266	0.42072	0.42072	0.4212	0.42386
<i>homogeneity</i>	0.9684	0.95641	0.95641	0.95656	0.96226
b) Jarak 2 <i>pixel</i>					
Sudut	0	45	90	135	rataan
<i>contrast</i>	0.21267	0.27583	0.17742	0.26307	0.23225
<i>correlation</i>	0.9621	0.95095	0.96838	0.95322	0.95866
<i>energy</i>	0.41459	0.40737	0.41598	0.40812	0.41152
<i>homogeneity</i>	0.94324	0.92903	0.94319	0.9288	0.93607
c) Jarak 3 <i>pixel</i>					
Sudut	0	45	90	135	rataan
<i>contrast</i>	0.31649	0.38297	0.24715	0.35969	0.32657
<i>correlation</i>	0.94366	0.93203	0.956	0.93616	0.94196
<i>energy</i>	0.40624	0.39882	0.40911	0.39983	0.4035
<i>homogeneity</i>	0.92591	0.91558	0.92854	0.91549	0.92138

Dari Tabel 4.2 didapatkan nilai statistika fitur tekstur citra berdasarkan perhitungan dari GLCMnya. Pada jarak *pixel* 1 dan pada arah sudut 0, 45, 90, dan 135 derajat, didapatkan nilai rataan *contrast* sebesar 0,11665 yang berarti pada jarak 1 *pixel* tersebut memiliki ukuran variasi antar derajat keabuan yang rendah atau

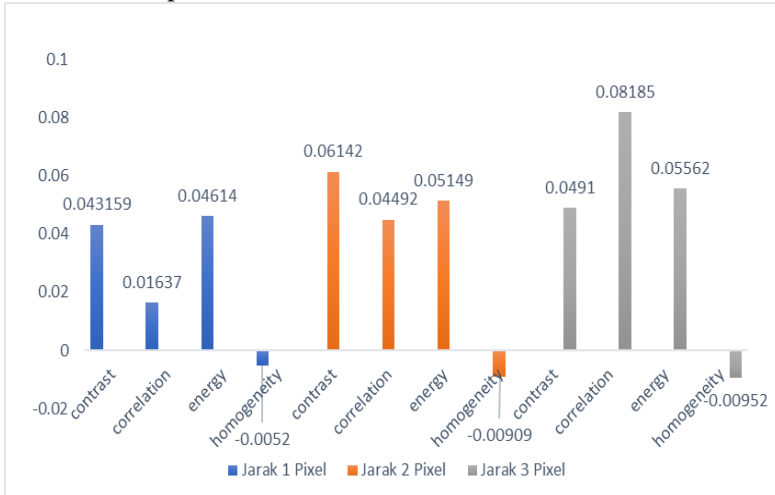
dapat dikatakan juga berdasarkan nilai tersebut letak intensitas *pixel-pixel*nya tidak terlalu jauh dari diagonal utama sehingga ukuran variasinya rendah. Pada jarak 2 *pixel* didapatkan nilai rata-rata *contrast* sebesar 0,23225, nilai tersebut menunjukkan ukuran variasi yang lebih besar terhadap diagonal utamanya dibandingkan jarak 1 *pixel*. Pada jarak 3 *pixel* didapatkan nilai rata-rata *contrast* sebesar 0,32657, terdapat peningkatan nilai *contrast* dibandingkan pada jarak 1 dan 2 *pixel*, sehingga terdapat peningkatan ukuran variasi pada diagonal utama untuk jarak 3 *pixel*.

Untuk nilai *correlation* pada jarak 1 *pixel* didapatkan nilai rata-rata sebesar 0,9792, hal ini menunjukkan pada jarak 1 *pixel* ukuran ketergantungan linear derajat keabuan citra cukup tinggi karena mendekati nilai 1. Pada jarak 2 *pixel* didapatkan nilai *correlation* sebesar 0,95866 terjadi penurunan nilai rata-rata *correlation* dapat dikatakan untuk jarak 2 *pixel* terjadi penurunan ketergantungan linear derajat keabuan citra, tetapi nilai yang didapatkan masih cukup tinggi sebab mendekati nilai 1, sehingga ketergantungan linear derajat keabuan tiap *pixel*-nya masih tinggi. Untuk *correlation* pada jarak 3 *pixel*, didapatkan nilai sebesar 0,94196 hal ini menunjukkan penurunan nilai ketergantungan linear antar *pixel-pixel*nya, tetapi nilainya masih cukup tinggi sebab mendekati nilai 1, sehingga masih memberikan petunjuk struktur linear dalam citra tersebut.

Nilai rata-rata *Energy* pada jarak 1 *pixel* didapatkan sebesar 0,42386, dapat dikatakan untuk jarak 1 *pixel* citra tersebut memiliki keteraturan yang cukup rendah, sebab nilainya cenderung mendekati nilai 0 atau dapat dikatakan *pixel-pixel* pada citra tersebut strukturnya tidak teratur atau bervariasi. Pada jarak 2 *pixel* didapatkan rata-rata nilai *energy* sebesar 0,41152 terjadi penurunan pada jarak 1 *pixel*, tetapi tidak terlalu jauh besarnya, dapat dikatakan pada jarak 2 *pixel*, citra tersebut juga memberikan struktur yang tidak teratur tetapi juga menurunkan nilai variasi *pixel*-nya. Pada jarak 3 *pixel* didapatkan rata-rata nilai *energy* sebesar 0,4035, dimana terjadi penurunan tingkat *energy*, nilai tersebut

memberikan kesimpulan pada jarak 3 *pixel*, stuktur citra tersebut memiliki struktur yang tidak teratur tetapi mengalami penurunan variasi dibandingkan pada jarak 1 *pixel* dan 2 *pixel*.

Nilai rataaan *homogeneity* untuk jarak 1 *pixel* didapatkan nilai sebesar 0,96226 yang berarti untuk jarak 1 *pixel* terdapat tingkat homogen yang tinggi sebab mendekati nilai 1, atau dapat di simpulkan juga pada jarak 1 *pixel* letak intensitasnya jauh terhadap diagonal utamanya. Pada jarak 2 *pixel* nilai rataaan *homogeneity*-nya didapatkan sebesar 0,93607 nilai tersebut mengalami penurunan dari jarak 1 *pixel*, tetapi nilai ini masih memberikan tingkat homogen yang tinggi antar *pixel*-nya. Pada jarak 3 *pixel* didapatkan nilai *homogeneity* sebesar 0,92138 nilai tersebut masih menunjukkan tingkat homogen yang tinggi antar *pixel-pixel* didalam citra tersebut. Setelah dilakukan perhitungan nilai statistika fitur tekstur citra, dilakukan perhitungan perubahan nilai statistika fitur tekstur citra sebelum tahapan *preprocessing* dan setelah *preprocessing*, yang di tampilkan didalam sebuah *clustered column chart* pada Gambar 4.15.



Gambar 4.15 *Clustered Column Chart* selisih rataaan Statistika Fitur Tekstur Citra sebelum dan setelah *preprocessing* data *ax T2 flair*

Berdasarkan Gambar 4.16 nilai *contrast* untuk jarak 1 *pixel*, 2 *pixel*, dan 3 *pixel* didapatkan nilai positif untuk tiap-tiap *pixel*-nya dapat disimpulkan dengan melakukan *preprocessing* data citra yang dihasilkan letak intensitas *pixel-pixel*nya semakin menjauhi diagonal utama sehingga mengecilkan variasinya. Nilai perubahan *contrast* ini tidak terlalu besar sehingga dapat dikatakan dengan melakukan *preprocessing* tidak terlalu merubah kualitas *contrast*-nya.

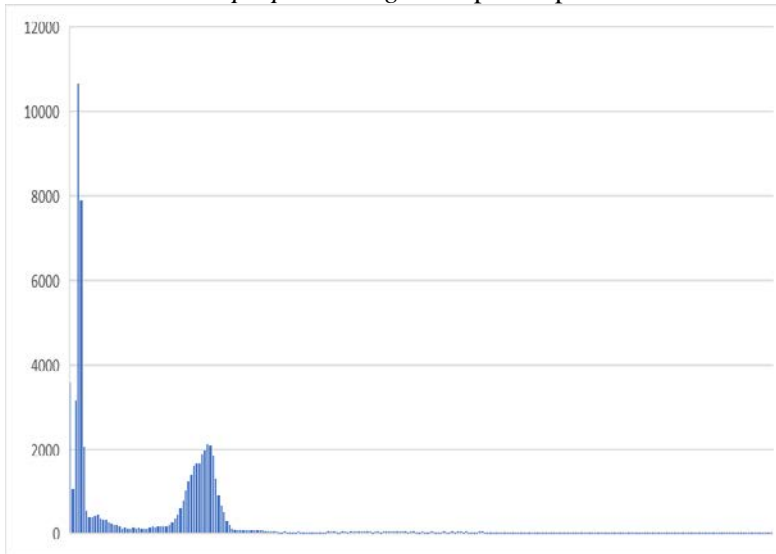
Untuk nilai *correlation* pada jarak 1 *pixel*, 2 *pixel*, dan 3 *pixel* didapatkan nilai positif untuk tiap-tiap *pixel*-nya ini mengakibatkan nilai ketergantungan linear derajat keabuan citra semakin tinggi, sehingga dengan melakukan *preprocessing* tersebut mengakibatkan kenaikan ketergantungan linear derajat keabuan citra di tiap-tiap jarak *pixel*-nya. Nilai perubahan *correlation* ini tidak terlalu besar di tiap-tiap *pixel*-nya, sehingga citra yang dihasilkan dapat dikatakan tidak merubah kualitas *correlation*-nya.

Untuk nilai *energy* pada jarak 1 *pixel*, 2 *pixel*, dan 3 *pixel* didapatkan nilai perubahan yang positif, dapat disimpulkan tiap-tiap jarak *pixel* pada citra tersebut keteraturannya cenderung meningkat atau terjadi peningkatan variasi di tiap-tiap intensitasnya. Peningkatan nilai *energy* tidak terlalu besar, sehingga dapat disimpulkan dengan melakukan *preprocessing* tidak terlalu merubah kualitas *energy*-nya.

Nilai *homogeneity* pada jarak 1 *pixel*, 2 *pixel*, dan 3 *pixel* didapatkan nilai negatif. Nilai negatif *homogeneity* tersebut, dapat disimpulkan setelah dilakukan *preprocessing* tiap-tiap jarak *pixel* tersebut terjadi penurunan tingkat homogen. Penurunan nilai *homogeneity* tidak terlalu besar sehingga dapat disimpulkan dengan melakukan *preprocessing* tidak terlalu merubah kualitas *homogeneity*-nya.

B. Analisis selanjutnya dilakukan terhadap citra MRI *sequence T1 memp + c* dengan mencari Histogram dan Statistika Fitur Citra-nya. Proses *preprocessing sequence ax T1 memp + C* dilakukan mengikuti tahapan pada Algoritma 4.1. Penentuan histogram dan

nilai statistika fitur citra berdasarkan citra sebelum di-*preprocessing* dan citra hasil dari *preprocessing*. Histogram citra sebelum dilakukan *preprocessing* ditampilkan pada Gambar 4.16.



Gambar 4.16 Histogram Citra Sebelum *preprocessing*

Gambar 4.16 menunjukkan histogram citra sebelum dilakukan *preprocessing* memiliki 2 puncak. Setelah dilakukan *preprocessing* dengan langkah-langkah yang sama seperti pada *preprocessing* data *ax T1 memp+C* data citra tersebut didapatkan histogram yang ditampilkan pada Gambar 4.17.



Gambar 4.17 Histogram Citra Setelah *preprocessing*

Setelah didapatkan histogram hasil *preprocessing* akan dihitung nilai statistika fitur tekstur citra sebelum dan setelah proses *preprocessing*. Nilai statistika fitur tekstur citra sebelum *preprocessing* ditampilkan pada Tabel 4.3.

Tabel 4.3 Nilai Statistika Fitur Tekstur Citra Sebelum *preprocessing* a) Jarak 1 Pixel, b) Jarak 2 Pixel, c) Jarak 3 Pixel

	a) Jarak 1 <i>pixel</i>				
Sudut	0	45	90	135	rataan
<i>contrast</i>	0.07258	0.11669	0.06890	0.11014	0.09208
<i>correlation</i>	0.94532	0.91223	0.94809	0.91716	0.9307
<i>energy</i>	0.41578	0.40634	0.41761	0.40731	0.41176
<i>homogeneity</i>	0.96908	0.95976	0.97121	0.96105	0.96528

b) Jarak 2 <i>pixel</i>					
Sudut	0	45	90	135	rataan
<i>contrast</i>	0.18815	0.29149	0.16963	0.27652	0.23145
<i>correlation</i>	0.8585	0.78153	0.87242	0.79257	0.8263
<i>energy</i>	0.39505	0.38105	0.39942	0.38203	0.38939
<i>homogeneity</i>	0.94717	0.93463	0.95206	0.93674	0.94265
c) Jarak 3 <i>pixel</i>					
Sudut	0	45	90	135	rataan
<i>contrast</i>	0.32754	0.48149	0.28029	0.45184	0.38529
<i>correlation</i>	0.75409	0.64037	0.78956	0.66252	0.71164
<i>energy</i>	0.37865	0.35999	0.38385	0.36105	0.37088
<i>homogeneity</i>	0.93101	0.91442	0.93727	0.91714	0.92496

Dari Tabel 4.3 didapatkan nilai statistika fitur tekstur citra berdasarkan perhitungan dari GLCMnya. Pada jarak *pixel* 1 dan pada arah sudut 0, 45, 90, dan 135 derajat, didapatkan nilai rataan *contrast* sebesar 0,09208 yang berarti pada jarak 1 *pixel* tersebut memiliki ukuran variasi antar derajat keabuan yang rendah atau dapat dikatakan juga berdasarkan nilai tersebut letak intensitas *pixel-pixel*nya tidak terlalu jauh dari diagonal utama sehingga ukuran variasinya rendah. Pada jarak 2 *pixel* didapatkan nilai rataan *contrast* sebesar 0,23145, nilai tersebut menunjukkan ukuran variasi yang lebih besar terhadap diagonal utamanya dibandingkan jarak 1 *pixel*. Pada jarak 3 *pixel* didapatkan nilai rataan *contrast* sebesar 0,38529, terdapat peningkatan nilai *contrast* dibandingkan pada jarak 1 dan 2 *pixel*, sehingga terdapat peningkatan ukuran variasi pada diagonal utama untuk jarak 3 *pixel*.

Untuk nilai *correlation* pada jarak 1 *pixel* didapatkan nilai rataan sebesar 0,9307, hal ini menunjukkan pada jarak 1 *pixel* ukuran ketergantungan linear derajat keabuan citra cukup tinggi karena mendekati nilai 1. Pada jarak 2 *pixel* didapatkan nilai

correlation sebesar 0,8263 terjadi penurunan nilai rata-ran *correlation* dapat dikatakan untuk jarak 2 *pixel* terjadi penurunan ketergantungan linear derajat keabuan citra, tetapi nilai yang didapatkan masih cukup tinggi sebab mendekati nilai 1, sehingga ketergantungan linear derajat keabuan tiap *pixel*-nya masih tinggi. Untuk *correlation* pada jarak 3 *pixel*, didapatkan nilai sebesar 0,071164 hal ini menunjukkan penurunan nilai ketergantungan linear antar *pixel-pixel*nya, tetapi nilainya masih cukup tinggi sebab mendekati nilai 1, sehingga masih memberikan petunjuk struktur linear dalam citra tersebut.

Nilai rata-ran *Energy* pada jarak 1 *pixel* didapatkan sebesar 0,41176, dapat dikatakan untuk jarak 1 *pixel* citra tersebut memiliki keteraturan yang cukup rendah, sebab nilainya cenderung mendekati nilai 0 atau dapat dikatakan *pixel-pixel* pada citra tersebut strukturnya tidak teratur atau bervariasi. Pada jarak 2 *pixel* didapatkan rata-ran nilai *energy* sebesar 0,38939 terjadi penurunan pada jarak 1 *pixel*, tetapi tidak terlalu jauh besarnya, dapat dikatakan pada jarak 2 *pixel*, citra tersebut juga memberikan struktur yang tidak teratur tetapi juga menurunkan nilai variasi *pixel*-nya. Pada jarak 3 *pixel* didapatkan rata-ran nilai *energy* sebesar 0,37088, dimana terjadi penurunan tingkat *energy*, nilai tersebut memberikan kesimpulan pada jarak 3 *pixel*, struktur citra tersebut memiliki struktur yang tidak teratur tetapi mengalami penurunan variasi dibandingkan pada jarak 1 *pixel* dan 2 *pixel*.

Nilai rata-ran *homogeneity* untuk jarak 1 *pixel* didapatkan nilai sebesar 0,96528 yang berarti untuk jarak 1 *pixel* terdapat tingkat homogen yang tinggi sebab mendekati nilai 1, atau dapat disimpulkan juga pada jarak 1 *pixel* letak intensitasnya jauh terhadap diagonal utamanya. Pada jarak 2 *pixel* nilai rata-ran *homogeneity*-nya didapatkan sebesar 0,94265 nilai tersebut mengalami penurunan dari jarak 1 *pixel*, tetapi nilai ini masih memberikan tingkat homogen yang tinggi antar *pixel*-nya. Pada jarak 3 *pixel* didapatkan nilai *homogeneity* sebesar 0,92496 nilai tersebut masih menunjukkan tingkat homogen yang tinggi antar *pixel-pixel* didalam citra tersebut. Setelah melakukan analisa statistika fitur

tekstur citra sebelum *preprocessing*, akan dilakukan juga perhitungan terhadap statistika fitur tekstur citra setelah *preprocessing* yang ditampilkan pada tabel 4.4.

Tabel 4.4 Nilai Statistika Fitur Tekstur Citra Setelah *preprocessing* a) Jarak 1 Pixel, b) Jarak 2 Pixel, c) Jarak 3 Pixel

a) Jarak 1 <i>pixel</i>					
Sudut	0	45	90	135	rataan
<i>contrast</i>	0.09895	0.13933	0.08872	0.13244	0.11486
<i>correlation</i>	0.97592	0.96615	0.97841	0.96782	0.9208
<i>energy</i>	0.46838	0.46179	0.4695	0.46225	0.46548
<i>homogeneity</i>	0.97407	0.96567	0.97421	0.9656	0.96989
b) Jarak 2 <i>pixel</i>					
Sudut	0	45	90	135	rataan
<i>contrast</i>	0.20322	0.25665	0.16326	0.24527	0.2171
<i>correlation</i>	0.95063	0.93784	0.96034	0.9406	0.94735
<i>energy</i>	0.45418	0.446	0.45887	0.44766	0.45168
<i>homogeneity</i>	0.95591	0.94856	0.96065	0.94965	0.9537
c) Jarak 3 <i>pixel</i>					
Sudut	0	45	90	135	rataan
<i>contrast</i>	0.28888	0.35842	0.22233	0.33995	0.30239
<i>correlation</i>	0.92993	0.93203	0.94607	0.91793	0.92685
<i>energy</i>	0.44384	0.43264	0.45052	0.43507	0.44052
<i>homogeneity</i>	0.9451	0.93634	0.95172	0.93817	0.94283

Dari Tabel 4.4 didapatkan nilai statistika fitur tekstur citra berdasarkan perhitungan dari GLCMnya. Pada jarak *pixel* 1 dan pada arah sudut 0, 45, 90, dan 135 derajat, didapatkan nilai rataan *contrast* sebesar 0,11486 yang berarti pada jarak 1 *pixel* tersebut memiliki ukuran variasi antar derajat keabuan yang rendah atau

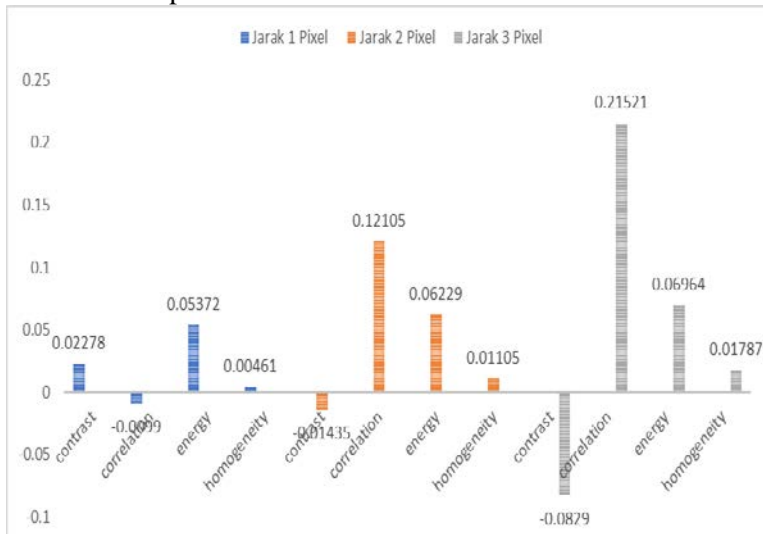
dapat dikatakan juga berdasarkan nilai tersebut letak intensitas *pixel-pixel*nya tidak terlalu jauh dari diagonal utama sehingga ukuran variasinya rendah. Pada jarak 2 *pixel* didapatkan nilai rata-rata *contrast* sebesar 0,2171, nilai tersebut menunjukkan ukuran variasi yang lebih besar terhadap diagonal utamanya dibandingkan jarak 1 *pixel*. Pada jarak 3 *pixel* didapatkan nilai rata-rata *contrast* sebesar 0.30239, terdapat peningkatan nilai *contrast* dibandingkan pada jarak 1 dan 2 *pixel*, sehingga terdapat peningkatan ukuran variasi pada diagonal utama untuk jarak 3 *pixel*.

Untuk nilai *correlation* pada jarak 1 *pixel* didapatkan nilai rata-rata sebesar 0.9208, hal ini menunjukkan pada jarak 1 *pixel* ukuran ketergantungan linear derajat keabuan citra cukup tinggi karena mendekati nilai 1. Pada jarak 2 *pixel* didapatkan nilai *correlation* sebesar 0.94735 terjadi penurunan nilai rata-rata *correlation* dapat dikatakan untuk jarak 2 *pixel* terjadi penurunan ketergantungan linear derajat keabuan citra, tetapi nilai yang didapatkan masih cukup tinggi sebab mendekati nilai 1, sehingga ketergantungan linear derajat keabuan tiap *pixel*-nya masih tinggi. Untuk *correlation* pada jarak 3 *pixel*, didapatkan nilai sebesar 0.92685 hal ini menunjukkan penurunan nilai ketergantungan linear antar *pixel-pixel*nya, tetapi nilainya masih cukup tinggi sebab mendekati nilai 1, sehingga masih memberikan petunjuk struktur linear dalam citra tersebut.

Nilai rata-rata *Energy* pada jarak 1 *pixel* didapatkan sebesar 0.46548 dapat dikatakan untuk jarak 1 *pixel* citra tersebut memiliki keteraturan yang cukup rendah, sebab nilainya cenderung mendekati nilai 0 atau dapat dikatakan *pixel-pixel* pada citra tersebut strukturnya tidak teratur atau bervariasi. Pada jarak 2 *pixel* didapatkan rata-rata nilai *energy* sebesar 0.45168 terjadi penurunan pada jarak 1 *pixel*, tetapi tidak terlalu jauh besarnya, dapat dikatakan pada jarak 2 *pixel*, citra tersebut juga memberikan struktur yang tidak teratur tetapi juga menurunkan nilai variasi *pixel*-nya. Pada jarak 3 *pixel* didapatkan rata-rata nilai *energy* sebesar 0.44052, dimana terjadi penurunan tingkat *energy*, nilai tersebut memberikan kesimpulan pada jarak 3 *pixel*, struktur citra tersebut

memiliki struktur yang tidak teratur tetapi mengalami penurunan variasi dibandingkan pada jarak 1 *pixel* dan 2 *pixel*.

Nilai rataaan *homogeneity* untuk jarak 1 *pixel* didapatkan nilai sebesar 0.96989 yang berarti untuk jarak 1 *pixel* terdapat tingkat homogen yang tinggi sebab mendekat nilai 1, atau dapat di simpulkan juga pada jarak 1 *pixel* letak intensitasnya jauh terhadap diagonal utamanya. Pada jarak 2 *pixel* nilai rataaan *homogeneity*-nya didapatkan sebesar 0.9537 nilai tersebut mengalami penurunan dari jarak 1 *pixel*, tetapi nilai ini masih memberikan tingkat homogen yang tinggi antar *pixel*-nya. Pada jarak 3 *pixel* didapatkan nilai *homogeneity* sebesar 0.94283 nilai tersebut masih menunjukkan tingkat homogen yang tinggi antar *pixel-pixel* didalam citra tersebut. Setelah dilakukan perhitungan nilai statistika fitur tekstur citra, dilakukan perhitungan perubahan nilai statistika fitur tekstur citra sebelum tahapan *preprocessing* dan setelah *preprocessing*, yang di tampilkan didalam sebuah *clustered column chart* pada Gambar 4.18.



Gambar 4.18 *Clustered Column Chart* selisih rataaan Statistika Fitur Tekstur Citra sebelum dan setelah *preprocessing* data ax T1 *memp+cS*

Berdasarkan Gambar 4.18 nilai *contrast* untuk jarak 1 *pixel* didapatkan nilai positif saat dilakukan *preprocessing* dapat disimpulkan dengan melakukan *preprocessing* data citra yang dihasilkan letak intensitas *pixel* semakin menjauhi diagonal utama pada jarak 1 *pixel* sehingga mengecilkan variasinya sedangkan didapatkan nilai negatif *contrast* pada jarak 2 *pixel* dan 3 *pixel*. Nilai negatif *contrast* ini menyimpulkan peningkatan variasi antar *pixel*nya.

Untuk nilai *correlation* pada jarak 1 *pixel* didapatkan nilai negatif untuk tiap-tiap *pixel*-nya ini mengakibatkan nilai ketergantungan linear derajat keabuan citra semakin turun pada jarak 1 *pixel* tetapi penurunan tersebut tidak terlalu besar sehingga tidak terlalu merubah kualitasnya pada jarak 1 *pixel*. Pada jarak 2 *pixel* dan 3 *pixel* terjadi peningkatan sehingga dengan melakukan *preprocessing* tersebut mengakibatkan peningkatan ketergantungan linear derajat keabuan citra di tiap-tiap jarak *pixel*-nya. Nilai perubahan *correlation* ini cukup besar untuk jarak 2 dan 3 *pixel*, sehingga citra yang dihasilkan dapat dikatakan merubah kualitas *correlation*-nya.

Untuk nilai *energy* pada jarak 1 *pixel*, 2 *pixel*, dan 3 *pixel* didapatkan nilai perubahan yang positif, dapat disimpulkan tiap-tiap jarak *pixel* pada citra tersebut keteraturannya cenderung meningkat atau terjadi peningkatan variasi di tiap-tiap intensitasnya. Peningkatan nilai *energy* cukup besar, sehingga dapat disimpulkan dengan melakukan *preprocessing* cukup merubah kualitas *energy*-nya.

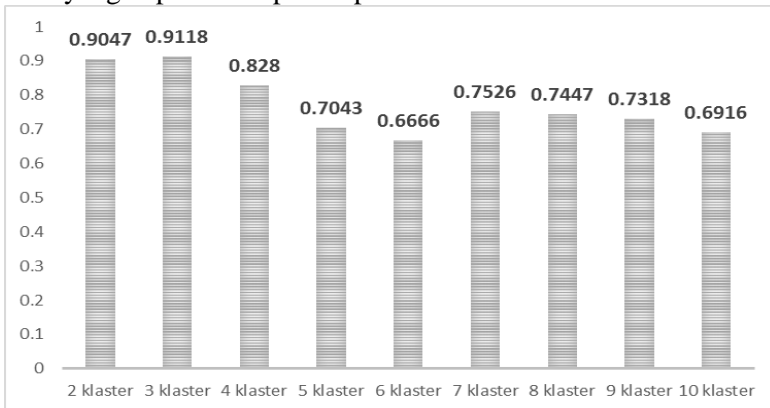
Nilai *homogeneity* pada jarak 1 *pixel*, 2 *pixel*, dan 3 *pixel* didapatkan nilai perubahan positif. Nilai positif *homogeneity* menyimpulkan setelah dilakukan *preprocessing* tiap-tiap jarak *pixel* tersebut terjadi peningkatan tingkat homogen. Peningkatan nilai *homogeneity* tiap-tiap *pixel* tidak terlalu besar sehingga dapat disimpulkan dengan melakukan *preprocessing* tidak terlalu merubah kualitas *homogeneity*-nya.

4.2 Uji coba, Pemilihan dan Validasi Jumlah kluster

Saat akan dilakukan segmentasi menggunakan metode klustering, akan dilakukan pemilihan jumlah kluster yang akan dipakai didalam proses segmentasi. Segmentasi menggunakan metode EM-GMM akan di uji coba menggunakan nilai *silhouette index* dan segmentasi metode FCM akan di uji coba menggunakan nilai PCI (*Partition Coefficient Index*). Nilai terbaik akan menjadi pertimbangan untuk memilih jumlah kluster, tetapi penilaian subjektif dari pihak medis penilaian utama dalam pemilihan jumlah klasternya.

4.2.1 Uji coba, Pemilihan dan Validasi Jumlah kluster EM-GMM

A. Validasi terhadap metode EM-GMM untuk segmentasi citra *ax T2 flair* adalah menggunakan nilai *silhouette index* uji coba dilakukan terhadap jumlah kluster 2-10 kluster dan dihasilkan nilai yang dapat di tampilkan pada Gambar 4.19.

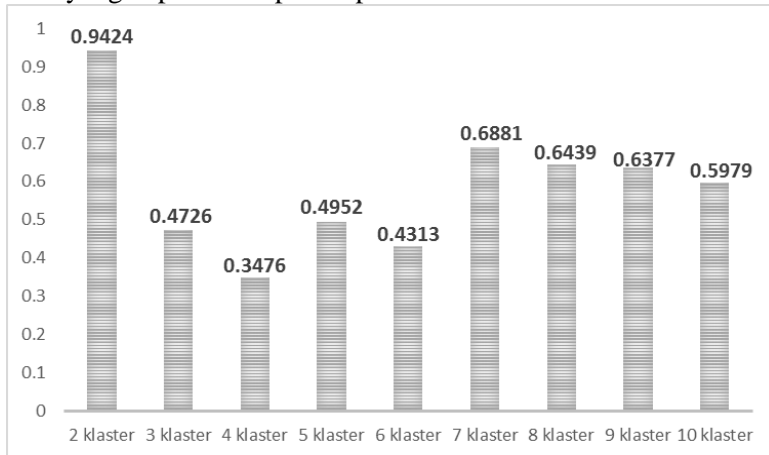


Gambar 4.19 Histogram Nilai *silhouette index ax T2 flair*

Berdasarkan Gambar 4.19, didapatkan nilai *silhouette index* tertinggi pada jumlah kluster 3 dengan nilai 0.9118. Selain dilakukan perhitungan nilai *silhouette index* akan dilakukan validasi subjektif secara medis untuk pemilihan jumlah kluster

tersebut. Didapatkan untuk pemilihan 3 kluster sudah cukup memenuhi penilaian secara subjektif secara medis.

B. Validasi terhadap metode EM-GMM untuk segmentasi citra *ax T1 memp+c* adalah menggunakan nilai *silhouette index* uji coba dilakukan terhadap jumlah kluster 2-10 kluster dan dihasilkan nilai yang dapat di tampilkan pada Gambar 4.20.



Gambar 4.20 Histogram Nilai *silhouette index* *ax T1 memp+c*

Berdasarkan Gambar 4.20, didapatkan nilai *silhouette index* tertinggi pada jumlah kluster 2 dengan nilai 0.9424. Validasi subjektif secara medis untuk pemilihan jumlah kluster tersebut tidak terpenuhi, sehingga dipilih nilai *silhouette index* tertinggi kedua yaitu 7 kluster. Untuk jumlah kluster 7 dapat di validasi subjektif secara medis, sehingga dipilih jumlah kluster 7 untuk data *ax T1 memp+c*.

C. Uji Normalitas Data *ax T2 Flair* menggunakan uji Kolmogorov-Smirnov. Data dari tiap-tiap kluster akan di uji apakah setiap kluster memiliki distribusi normal. Dengan menggunakan bantuan aplikasi MINITAB hasil uji Kolmogorov-Smirnov dapat di tampilkan di dalam Tabel 4.5

\ Tabel 4.5 Uji Kolgomorov Smirnov data tiap-tiap kelas *ax T2 Flair*

klaster	P-value	KS
1	<0,01	0,516
2	<0,01	0,045
3	<0,01	0,042

Berdasarkan Tabel 4.4 didapat semua data di tiap-tiap kelas tidak memiliki distribusi normal sebab nilai *p-value* kurang dari nilai alpha. Namun, didalam penelitian ini, data tiap-tiap kelas diasumsikan memiliki distribusi normal, sehingga dapat dilanjutkan untuk dilakukan analisis klaster menggunakan EM-GMM.

D. Uji Normalitas Data *ax T1 memp+c* menggunakan uji Kolmogorov-Smirnov. Data dari tiap-tiap klaster akan di uji apakah setiap klaster memiliki distribusi normal. Dengan menggunakan bantuan aplikasi MINITAB hasil uji Kolmogorov-Smirnov dapat di tampilkan di dalam Tabel 4.6

Tabel 4.6 Uji Kolgomorov Smirnov data tiap-tiap kelas *ax T1 memp+c*

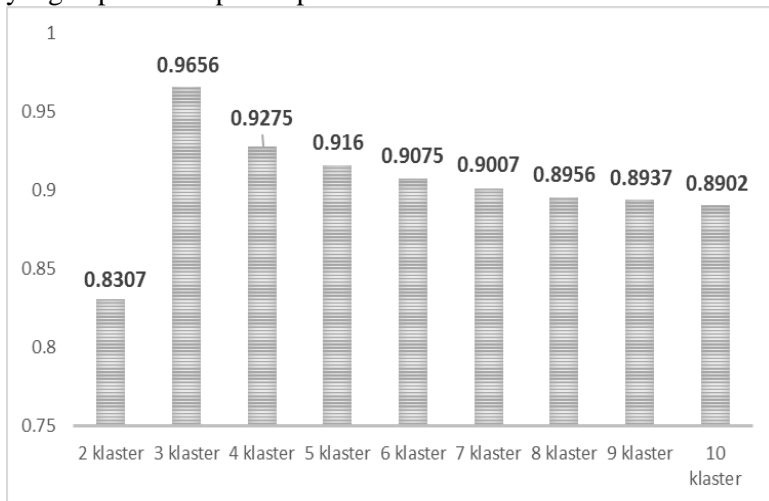
klaster	P-value	KS
1	<0,01	0,518
2	<0,01	0,182
3	<0,01	0,108
4	<0,01	0,112
5	<0,01	0,136
6	<0,01	0,110
7	0,018	0,181

Berdasarkan Tabel 4.6 didapat semua data di tiap-tiap kelas tidak memiliki distribusi normal sebab nilai *p-value* kurang dari nilai alpha. Namun, didalam penelitian ini, data tiap-tiap kelas diasumsikan memiliki distribusi normal, sehingga dapat

dilanjutkan untuk dilakukan analisis kluster menggunakan EM-GMM.

4.2.2 Uji coba, Pemilihan dan Validasi Jumlah kluster FCM

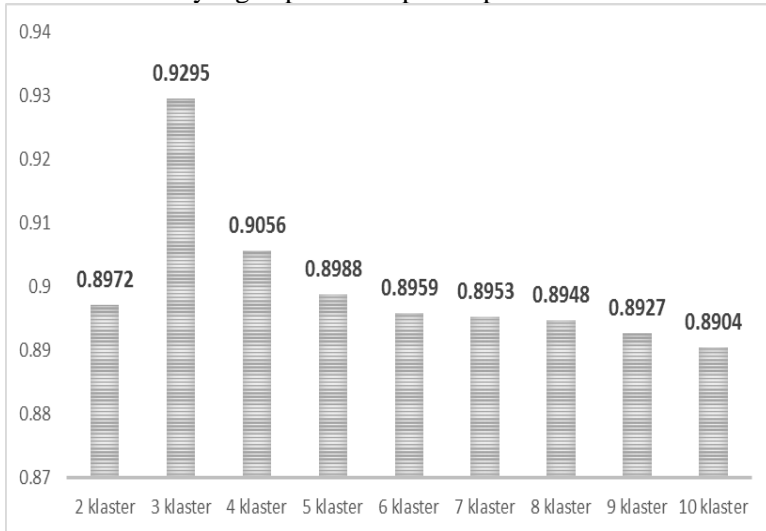
A. Validasi terhadap metode FCM untuk segmentasi citra *ax T2 flair* adalah menggunakan nilai *partition coefficient index* uji coba dilakukan terhadap jumlah kluster 2-10 kluster. Perhitungan menggunakan bantuan aplikasi excel dan minitab dihasilkan nilai yang dapat di tampilkan pada Gambar 4.21.



Gambar 4.21 Histogram Nilai *partition coefficient index* *ax T2 flair*

Berdasarkan Gambar 4.21, didapatkan nilai *partition coefficient index* tertinggi pada jumlah kluster 3 dengan nilai 0.9656. Selain dilakukan perhitungan nilai *partition coefficient index* akan dilakukan validasi subjektif secara medis untuk pemilihan jumlah kluster tersebut. Didapatkan untuk pemilihan 3 kluster sudah cukup memenuhi penilaian secara subjektif secara medis.

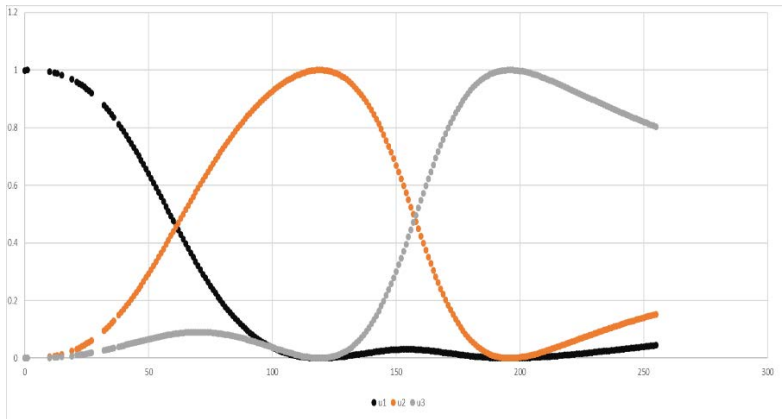
B. Validasi terhadap metode FCM untuk segmentasi citra *ax T1 memp+c* adalah menggunakan nilai *partition coefficient index* uji coba dilakukan terhadap jumlah kluster 2-10 kluster dan dihasilkan nilai yang dapat di tampilkan pada Gambar 4.22.



Gambar 4.22 Histogram Nilai *partition coefficient index* *ax T1 memp+c*

Berdasarkan Gambar 4.22, didapatkan nilai *partition coefficient index* tertinggi pada jumlah kluster 3 dengan nilai 0.9295. Validasi subjektif secara medis untuk pemilihan jumlah kluster tersebut tidak terpenuhi, sehingga dipilih nilai *partition coefficient index* lainnya sampai didapatkan jumlah kluster yang tervalidasi. Untuk jumlah kluster 6 dapat di validasi subjektif secara medis, sehingga dipilih jumlah kluster 6 untuk data *ax T1 memp+c*.

C. Dilakukan uji coba untuk mengetahui bentuk fungsi keanggotaan pada data *ax T2 Flair*. Didapatkan hasil pada gambar 4.24. Berdasarkan gambar tersebut dengan memakai 3 kluster, didapatkan fungsi keanggotaan memiliki bentuk tidak simetris.



Gambar 4.23 Uji Coba Fungsi Keanggotaan *fuzzy c-means* 3 kluster

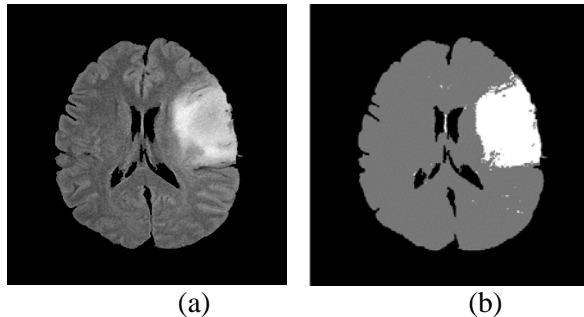
4.3 Segmentasi Citra dengan Metode Klustering

Citra hasil *preprocessing* akan disegmetasi dengan metode klustering. Yaitu menggunakan metode *Expectation Maximization-Gaussian Mixture Model* dan *Fuzzy C-Means*. Data yang diuji klasterig adalah *sequence ax T2 flair* dan *sequence ax T1 memp+c*. Berdasarkan pandangan medis segmentasi data *sequence ax T2 flair* dilakukan untuk menentukan area pembengkakan akibat tumor dan segmentasi *sequence ax T1 memp+c* dilakukan untuk menentukan area tumornya. Pemilihan jumlah kluster dan pemilihan *slice* citra yang dipakai untuk analisis adalah berdasarkan pandangan subjektif secara medis. Analisis akan dilakukan pada *slice* ke 18 citra setentang *sinus maxillaris* yang terindikasi adanya pembengkakan tumor dan tumor.

4.3.1 Segmentasi Expectation Maximization-Gaussian Mixture Model *ax T2 flair*

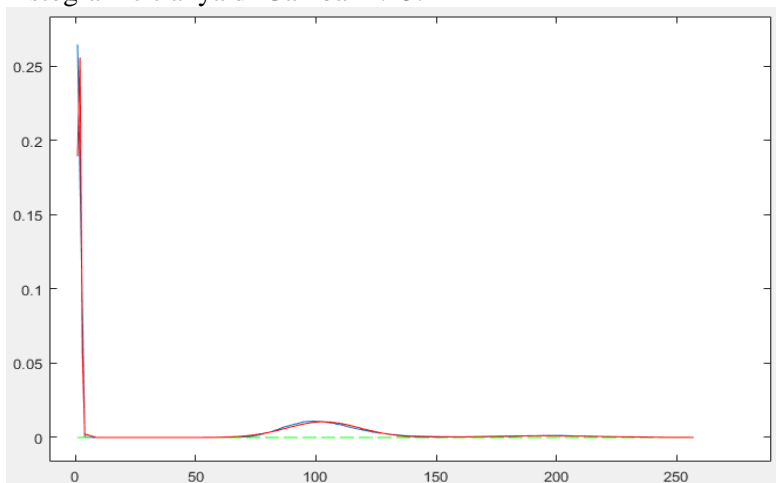
A. Citra hasil *preprocessing* akan disegmentasi dengan metode klustering. Metode yang pertama adalah *Expectation Maximization-Gaussian Mixture Model*. Dengan mengasumsikan data berdistribusi normal, pada data pertama yaitu *sequence ax T2 flair* akan di segmentasi untuk dicari area pembengkakannya. Uji

coba dilakukan segmentasi menggunakan 3 kluster. Citra segmentasi dapat ditampilkan pada Gambar 4.24.



Gambar 4.24 Citra hasil Segmentasi 3 kluster, a) Citra Hasil *preprocessing* b) Citra hasil segmentasi EM-GMM

Berdasarkan Gambar 4.24 (b) didapatkan hasil segmentasi metode klustering EM-GMM dengan 3 buah kluster, warna putih merupakan area kluster yang diduga sebagai area pembengkakan tumor. Selain itu akan dilihat bentuk distribusi dan berdasarkan histogram citranya di Gambar 4.25.



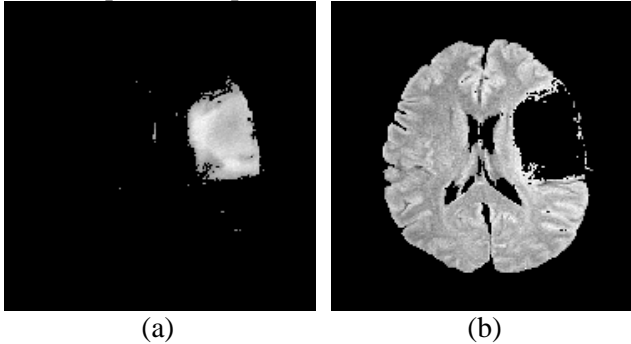
Gambar 4.25 Histogram Citra dan distribusi EM-GMM *ax T2 flair*

Berdasarkan Gambar 4.25 didapatkan 3 puncak berdistribusi normal dengan bantuan program dapat dimodelkan menjadi:

$$\begin{aligned} f(x|\mu_j, \sigma_j^2, p_j) &= \sum_{j=1}^3 p_j N(x|\mu_j, \sigma_j^2) = p_1 N(x|\mu_1, \sigma_1^2) + p_2 N(x|\mu_2, \sigma_2^2) + p_3 N(x|\mu_3, \sigma_3^2) \\ &= 0,5296N(x|1,668;0,5584) + 0,4004N(x|103,1665;22) \quad (4.1) \\ &\quad + 0,0699N(x|194,0287;600,1852) \end{aligned}$$

Berdasarkan persamaan (4.1) didapatkan masing-masing distribusi yang mewakili kelas-kelasnya. Dengan analisa secara subjektif berdasarkan bantuan medis, pada area $N(x|194,0286;600,1852)$ merupakan area pembengkakan tumor. Area ini yang menjadi fokus perhatian analisis atau disebut ROI (*Region of Interest*). Area $N(x|103,1664;229,51)$ merupakan area otak yang bukan pembengkakan tumor dan area $N(x|1,668;0,5584)$ adalah area diluar kepala. Kedua daerah ini merupakan area Non-ROI (*Non-Region of Interest*).

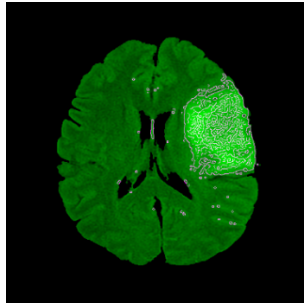
metode klustering yang telah dibuat, dapat dipisahkan menjadi area ROI (*Region of Interest*) dan Non-ROI (*Non-Region of Interest*), dapat dilihat pada Gambar 4.26.



Gambar 4.26 Citra ROI dan Non-ROI segmentasi klustering EM-GMM,
a) Area ROI, b) Area Non-ROI

Berdasarkan Gambar 4.26 (a) didapatkan area ROI, yaitu area otak yang diduga sebagai area pembengkakan area tumor dan

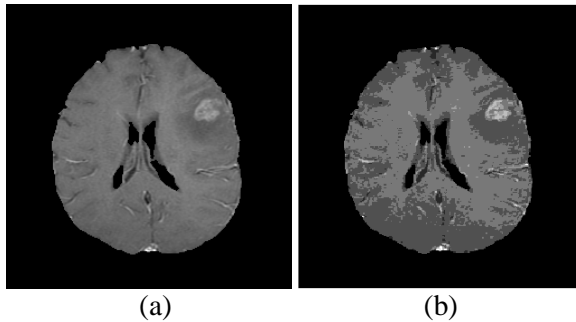
Gambar 4.26 (b) merupakan area Non-ROI yaitu area otak yang bukan merupakan area dugaan pembengkakan dari tumor. Berdasarkan citra yang didapatkan masih terdapat area-area kecil yang berjauhan area-area kecil merupakan *noise*. Selain itu dilakukan *subjective analysis* dengan mendeteksi tepi pada perubahan intensitas pada ROI, didapatkan hasil pada Gambar 4.27.



Gambar 4.27 *Subjective Analysis* Region of Interest sequence ax T2 *flair*

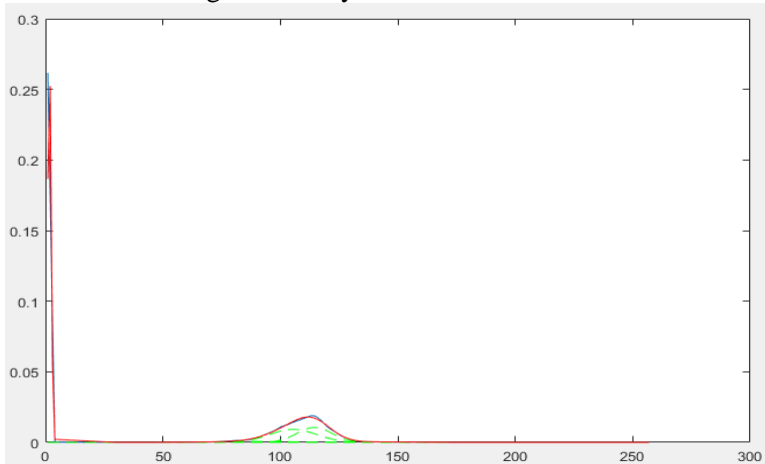
Berdasarkan Gambar 4.27 dilakukan *Subjective Analysis* pada *Region of Interest* dengan cara mendeteksi tepi pada area ROI-nya. Tujuannya adalah memberikan penilaian secara subjektif terhadap area pembengkakan tumor pada citra aslinya. Selain itu terdeteksi banyak garis-garis pada area dugaan tumor, garis-garis ini merupakan deteksi area korteks otak besar.

B. Dengan menggunakan Metode Expectation Maximization-Gaussian Mixture Model akan melakukan segmentasi terhadap data ax T1 memp + C. Dengan mengasumsikan data berdistribusi normal, data ax T1 memp + C akan disegmentasi untuk ditentukan area tumor otak pada citra tersebut. Uji coba dilakukan segmentasi menggunakan 7 klaster. Citra hasil segmentasi dapat ditampilkan pada Gambar 4.28.



Gambar 4.28 Citra hasil Segmentasi 7 klaster, a) Citra Hasil *preprocessing*, b) Citra hasil segmentasi EM-GMM

Berdasarkan Gambar 4.28 (b) didapatkan hasil segmentasi dengan metode klustering EM-GMM didapatkan 7 buah klaster, warna putih merupakan area klaster yang diduga sebagai area pembengkakan tumor. Selain itu akan dilihat bentuk distribusi dan berdasarkan histogram citranya di Gambar 4.29.



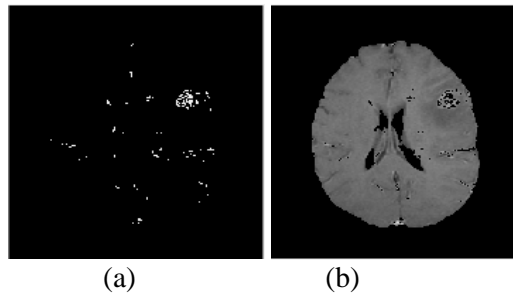
Gambar 4.29 Histogram Citra dan distribusi EM-GMM *ax T1 memp+c*

Berdasarkan Gambar 4.29 didapatkan 7 puncak berdistribusi normal dengan bantuan program dapat dimodelkan menjadi:

$$\begin{aligned}
f(x|\mu_j, \sigma_j^2, p_j) &= \\
\sum_{j=1}^7 p_j N(x|\mu_j, \sigma_j^2) &= p_1 N(x|\mu_1, \sigma_1^2) + p_2 N(x|\mu_2, \sigma_2^2) + \\
&\quad p_3 N(x|\mu_3, \sigma_3^2) + p_4 N(x|\mu_4, \sigma_4^2) + \\
&\quad p_5 N(x|\mu_5, \sigma_5^2) + p_6 N(x|\mu_6, \sigma_6^2) + \\
&\quad p_7 N(x|\mu_7, \sigma_7^2) \\
&= 0,5218 N(x|1,6688; 0,5594) + \tag{4.2} \\
&\quad 0,030 N(x|89,9493; 284,2501) + \\
&\quad 0,2305 N(x|105,4533; 99,1893) + \\
&\quad 0,1969 N(x|114,5002; 53,8884) + \\
&\quad 0,0142 N(x|127,6747; 213,7550) + \\
&\quad 0,0040 N(x|147.3737; 134,5424) + \\
&\quad 0,0020 N(x|171,6294; 1190,0888)
\end{aligned}$$

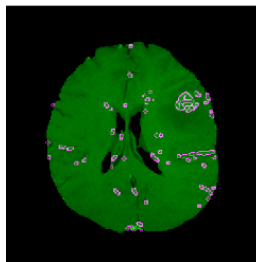
Berdasarkan persamaan (4.2) didapatkan masing-masing distribusi yang mewakilkan kelas-kelasnya. Dengan analisa secara subjektif berdasarkan bantuan medis, pada area $N(x|127,6747;213,7550)$ merupakan area tumor. Area ini yang menjadi fokus perhatian analisis atau disebut ROI (*Region of Interest*). Area $N(x|89,9493;284,2501)$, $N(x|105,4533;99,1893)$, $N(x|114,5002;53,8884)$, $N(x|147,3737;134,5424)$, dan $N(x|171,6294;1190,0888)$ merupakan area otak yang bukan tumor dan area $N(x|1,6688;0,5594)$ adalah area diluar kepala. Keenam daerah ini merupakan area Non-ROI (*Non-Region of Interest*).

Berdasarkan metode klastering yang telah dibuat, dapat dipisahkan menjadi area ROI (*Region of Interest*) dan Non-ROI (*Non-Region of Interest*), dapat dilihat pada Gambar 4.29.



Gambar 4.30 Citra ROI dan Non-ROI segmentasi klastering EM-GMM, a) Area ROI, b) Area Non-ROI

Berdasarkan Gambar 4.30 (a) didapatkan area ROI, yaitu area otak yang diduga sebagai area area tumor dan Gambar 4.29 (b) merupakan area Non-ROI yaitu area otak yang bukan merupakan area dugaan tumor. Berdasarkan citra yang didapatkan masih terdapat area-area kecil yang berjauhan area-area kecil merupakan *noise*. Selain itu dilakukan *subjective analysis* dengan mendeteksi tepi pada perubahan intensitas pada ROI, didapatkan hasil pada Gambar 4.31.

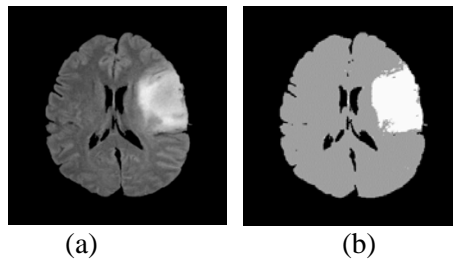


Gambar 4.31 *Subjective Analysis* Region of Interest sequence ax T1 *memp + C*

Berdasarkan Gambar 4.31 dilakukan *Subjective Analysis* pada *Region of Interest* dengan cara mendeteksi tepi pada area ROI-nya. Tujuannya adalah menampilkan area tumor pada citra aslinya, sehingga dapat lebih jelas untuk menentukan area tumornya secara medis.

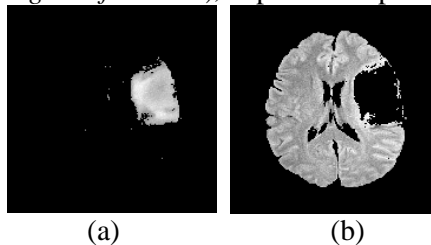
4.3.2 Segmentasi Fuzzy-C Means sequence ax T2 flair

A. Citra hasil *preprocessing* akan disegmentasi dengan metode klastering. Metode yang dipakai adalah *Fuzzy C-Means*. Pada data pertama yaitu *sequence ax T2 flair* akan di segmentasi untuk dicari area pembengkakannya. Uji coba dilakukan segmentasi menggunakan 3 klaster. Citra segmentasi dapat ditampilkan pada Gambar 4.32



Gambar 4.32 Citra hasil Segmentasi 3 klaster, a) Citra Hasil *preprocessing* b) Citra hasil segmentasi FCM

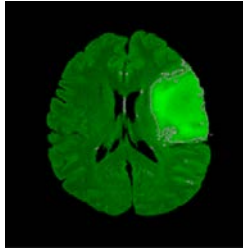
Berdasarkan Gambar 4.31 (b) didapatkan hasil segmentasi dengan metode klastering FCM didapatkan 3 buah klaster, warna putih merupakan area klaster yang diduga sebagai area pembengkakan tumor. Berdasarkan metode klastering yang telah dibuat, dapat dipisahkan menjadi area *ROI (Region of Interest)* dan *Non-ROI (Non-Region of Interest)*, dapat dilihat pada Gambar 4.32.



Gambar 4.33 Citra ROI dan Non-ROI segmentasi klastering FCM, a) Area ROI, b) Area Non-ROI

Berdasarkan Gambar 4.33 (a) didapatkan area ROI, yaitu area otak yang diduga sebagai area pembengkakan area tumor dan Gambar 4.32 (b) merupakan area Non-ROI yaitu area otak yang

bukan merupakan area dugaan pembengkakan dari tumor. Berdasarkan citra yang didapatkan masih terdapat area-area kecil yang berjauhan area-area kecil merupakan *noise*. Selain itu dilakukan *subjective analysis* dengan mendeteksi tepi pada perubahan intensitas pada ROI, didapatkan hasil pada Gambar 4.33.



Gambar 4.34 *Subjective Analysis Region of Interest sequence ax T2 flair*

Berdasarkan Gambar 4.34 dilakukan *Subjective Analysis* pada *Region of Interest* dengan cara mendeteksi tepi pada area ROI-nya. Tujuannya adalah memberikan penilaian secara subjektif terhadap area pembengkakan tumor pada citra aslinya. Pusat klaster dari segmentasi tersebut dapat dilihat didalam Tabel 4.7

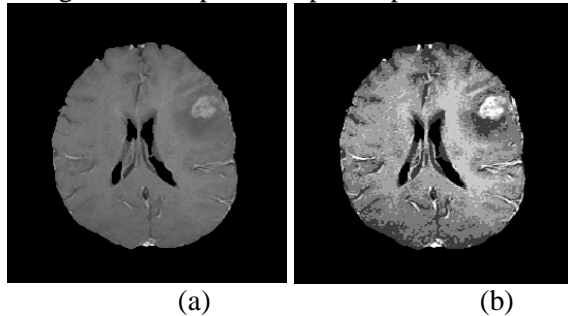
Tabel 4.7 Nilai Pusat klaster FCM *sequence ax T2 flair*

Pusat Klaster 1	Pusat Klaster 2	Pusat Klaster 3
0.0689	101.8628	196.1618

Tiga buah pusat klaster mewakili masing-masing area dari otak. Untuk pusat klaster 0.0689 merupakan area hitam diluar kepala dan pusat klaster 101.8628 merupakan area kepala non pembengkakan tumor. Kedua daerah ini merupakan area Non-ROI. Pusat klaster ke tiga dengan pusat klaster 196.1618 merupakan area pembengkakan tumor area ini merupakan area ROI.

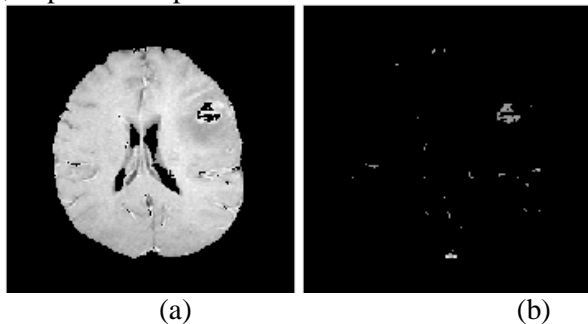
B. Dengan menggunakan Metode *Fuzzy C-Means* akan dilakukan segmentasi terhadap data *ax T1 memp* + C. Data *ax T1 memp* + C akan disegmentasi untuk ditentukan area tumor otak

pada citra tersebut. Uji coba dilakukan segmentasi menggunakan 6 klaster. Citra segmentasi dapat ditampilkan pada Gambar 4.34.



Gambar 4.35 Citra hasil Segmentasi 6 klaster, a) Citra Hasil *preprocessing* b) Citra hasil segmentasi FCM

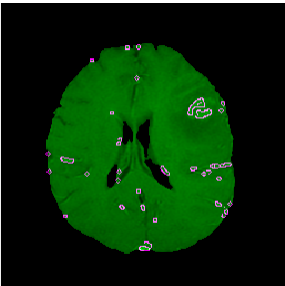
Berdasarkan Gambar 4.35 (b) didapatkan hasil segmentasi dengan metode klastering FCM didapatkan 6 buah klaster, warna putih merupakan area klaster yang diduga sebagai area tumor. Berdasarkan metode klastering yang telah dibuat, dapat dipisahkan menjadi area *ROI (Region of Interest)* dan *Non-ROI (Non-Region of Interest)*, dapat dilihat pada Gambar 4.35.



Gambar 4.36 Citra ROI dan Non-ROI segmentasi klastering EM-GMM, a) Area ROI, b) Area Non-ROI

Berdasarkan Gambar 4.36 (a) didapatkan area ROI, yaitu area otak yang diduga sebagai area area tumor dan Gambar 4.35 (b) merupakan area Non-ROI yaitu area otak yang bukan merupakan area dugaan tumor. Berdasarkan citra yang didapatkan

masih terdapat area-area kecil yang berjauhan area-area kecil merupakan *noise*. Selain itu dilakukan *subjective analysis* dengan mendeteksi tepi pada perubahan intensitas pada ROI, didapatkan hasil pada Gambar 4.37.



Gambar 4.37 *Subjective Analysis* Region of Interest sequence ax T1 *memp+c*

Berdasarkan Gambar 4.37 dilakukan *Subjective Analysis* pada *Region of Interest* dengan cara mendeteksi tepi pada area ROI-nya. Tujuannya adalah memberikan penilaian secara subjektif terhadap area pembengkakan tumor pada citra aslinya. Pusat klaster dari segmentasi tersebut dapat dilihat didalam Tabel 4.7

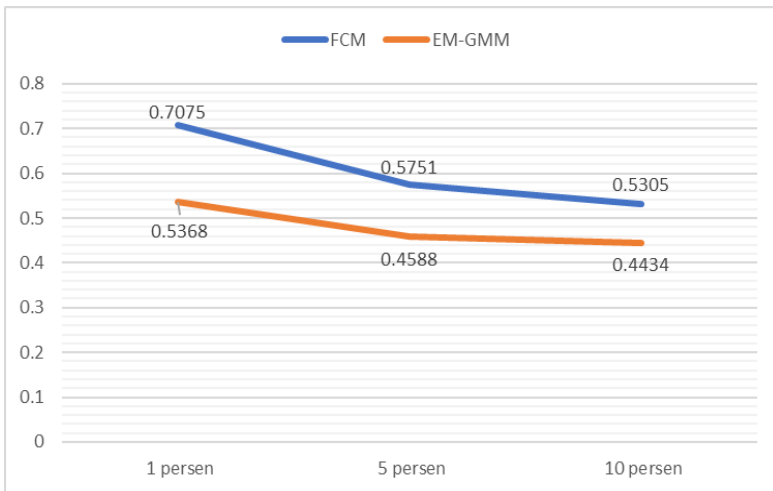
Tabel 4.7 Nilai Pusat klaster FCM *sequence ax T2 flair*

Pusat Klaster 1	Pusat Klaster 2	Pusat Klaster 3	Pusat Klaster 4	Pusat Klaster 5	Pusat Klaster 6
0.0084	90.2681	104.2329	114.3431	124.4859	151.2782

Enam buah pusat klaster mewakili masing-masing area dari otak. Untuk pusat klaster 0.0084 merupakan area hitam diluar kepala dan pusat klaster 90.2681, 104.2329, 114.3431, dan 151.2782 merupakan area kepala non tumor. Kedua daerah ini merupakan area Non-ROI. Pusat klaster ke lima dengan pusat klaster 124.4859 merupakan area tumor area ini merupakan area ROI.

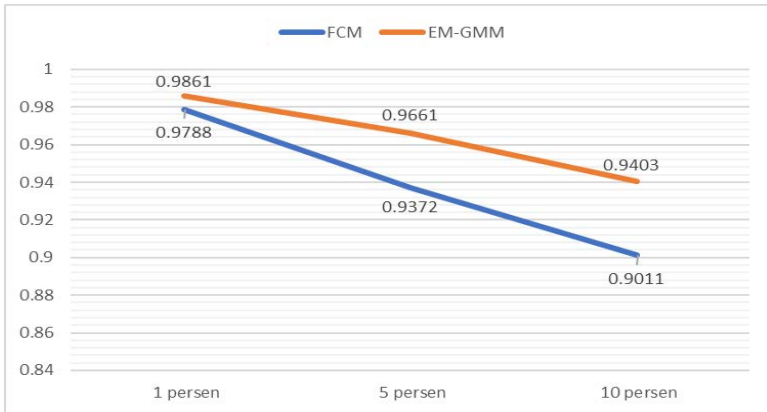
4.4 Similarity

Akan dilakukan pengujian tingkat *similarity* tiap-tiap metode berdasarkan tingkat *noise* 1%, 5%, dan 10% menggunakan *gaussian noise* dan *salt and pepper noise*. Nilai tersebut merupakan nilai rata-rata *similarity* untuk klastering 2-10 klaster. Untuk data *ax T2 flair* didapatkan nilai *similarity* yang ditampilkan pada Gambar 4.38.



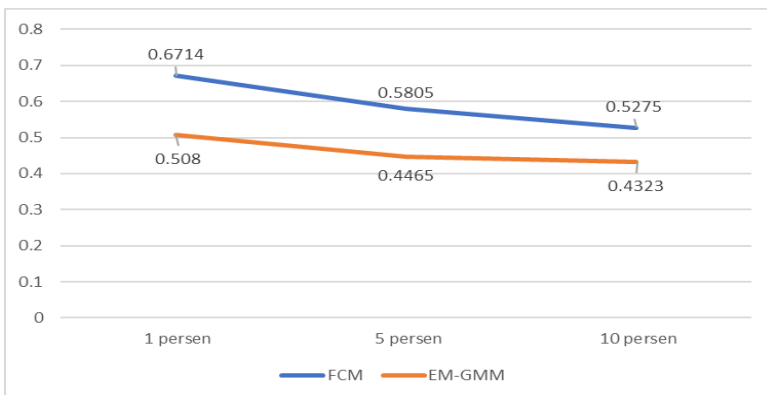
Gambar 4.38 *Similarity* pada data *ax T2 flair* dengan *Gaussian Noise*

Berdasarkan Gambar 4.38 didapatkan nilai *similarity* dengan pemberian *gaussian noise*. Nilai *similarity* yang didapatkan menunjukkan metode FCM memberikan hasil yang lebih bagus di tiap-tiap tingkatan *noise* nya. Sehingga dapat disimpulkan untuk pemberian *gaussian noise* metode FCM dominan lebih *robust* daripada metode EM-GMM. Setelah itu akan dibandingkan nilai *similarity* menggunakan *salt and pepper noise* ditampilkan pada Gambar 4.39.



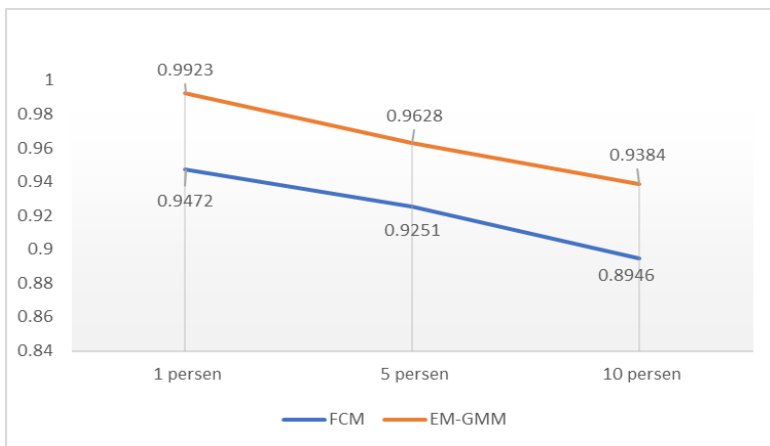
Gambar 4.39 *Similarity* pada data ax T2 *flair* dengan *Salt and Pepper Noise*

Berdasarkan Gambar 4.39 nilai *similarity* metode EM-GMM didapatkan lebih unggul dibanding metode FCM. Dapat disimpulkan dengan pemberian *salt and pepper noise*, metode EM-GMM akan lebih *robust* dibandingkan metode FCM. Setelah itu akan dilakukan perhitungan nilai *similarity* pada data ax T1 *memp+c*. Nilai *similarity* ditampilkan pada Gambar 4.40.



Gambar 4.40 *Similarity* pada data ax T1 *memp+c* dengan *gaussian noise*

Berdasarkan Gambar 4.40 didapatkan nilai *similarity* dengan pemberian *gaussian noise*. Nilai *similarity* yang didapatkan menunjukkan metode FCM memberikan hasil yang lebih bagus di tiap-tiap tingkatan *noise* nya. Sehingga dapat disimpulkan untuk pemberian *gaussian noise* metode FCM dominan lebih *robust* daripada metode EM-GMM. Setelah itu akan dibandingkan nilai *similarity* menggunakan *salt and pepper noise* ditampilkan pada Gambar 4.41.



Gambar 4.41 *Similarity* pada data *ax T1 memp+c* dengan *Salt and Pepper Noise*

Berdasarkan Gambar 4.41 nilai *similarity* metode EM-GMM didapatkan lebih unggul dibanding metode FCM. Dapat disimpulkan dengan pemberian *salt and pepper noise*, metode EM-GMM akan lebih *robust* dibandingkan metode FCM. Setelah itu akan dilakukan perhitungan nilai *similarity* pada data *ax T1 memp+c*.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah dilakukan implementasi, uji coba, analisis dan pembahasan pada Bab IV, maka berikut merupakan kesimpulan-kesimpulan yang didapatkan dari penelitian ini berkaitan dengan latar belakang dan tujuan penelitian.

Terjadi perubahan kualitas citra berdasarkan perubahan nilai Statistika fitur citranya. Untuk data *ax T2 flair* tidak ada perubahan yang terlalu besar pada nilai *contrast*, *correlation*, *energy*, dan *homogeneity* sehingga untuk data sebelum dan sesudah preprocessing tidak mengalami perubahan kualitas. Untuk data *ax T1 memp+C* terjadi peningkatan nilai *correlation* pada jarak 2 *pixel* dan 3 *pixel* yang cukup tinggi, hal ini mengakibatkan terdapat peningkatan ketergantungan linear derajat keabuan di tiap-tiap jarak *pixel* pada citra setelah *preprocessing*.

Lalu dengan menggunakan menggunakan validasi kluster data *ax T2 flair* didapatkan jumlah kluster 3 untuk metode EM-GMM dan FCM, sedangkan untuk data *ax T1 memp+c* menghasilkan hasil yang tidak valid, sehingga pemilihan jumlah kluster berdasarkan keputusan subjektif secara medis, yaitu menggunakan 7 kluster untuk metode EM-GMM dan 6 kluster untuk metode FCM. Nilai similarity yang dihasilkan memberikan kesimpulan metode EM-GMM lebih robust terhadap *salt and pepper noise* ketimbang FCM. Dan metode FCM lebih robust terhadap *gaussian noise* ketimbang EM-GMM.

5.2 Saran

Berdasarkan analisis yang telah dilakukan pada penelitian ini, beberapa saran untuk penelitian selanjutnya adalah sebagai berikut.

1. Memakai metode yang melibatkan metode spasial sehingga *pixel-pixel* yang sekiranya berada jauh dari ROI bisa dikeluarkan dari analisis.

2. Untuk EM-GMM diharapkan menggunakan metode *mixture* yang bentuknya mengikuti pola distribusi pada citra yang ada.
3. Menggunakan IDE lainnya yang lebih cepat dan ringan saat implementasi
4. Membandingkan dengan metode *hard clustering*
5. Menganalisa *sequence* lainnya di kepala sehingga memberikan visualisasi tumor yang lebih akurat.

DAFTAR PUSTAKA

- Balafar, M. A. (2013). *Fast and Robust Gaussian Mixture Model for MRI Brain Image Segmentation*. International
- Bezdek, J. C. (1975). *Mathematical Models for Systematic and Taxonomy*. Eight International Conference on Numerical Taxonomy, San Fransisco, CA, hal 143-165
- Bezdek, J. C. (1981) *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York Plenum.
- Bhagwat, K., More, D., Shinde, S., Daga, A., dan Tornekar, R. (2013). *Comparative Study of Brain Tumour Detection Using K-Means, Fuzzy C Means, and Hierarchical Clustering Algorithms*. International Journal of Scientific & Engineering Research, Volume 4, Issue 6
- Bhide, A. S., Patil, P., Shraddha, D. (2012). *Brain Segmentation using Fuzzy C Means Clustering to detect Tumour Region*. International Journal of Advanced Research in Computer Science and Electronic Engineering.
- Dempster, A. P., Laird, N. M., dan Rubin, D. B. (1997). *Maximum Likelihood from Incomplete Data via EM Algoritm*. Journal of the Royal Statistical Society. Series B (Methodological), Vol. 39. 1.
- Dorigo, M (1992). *Optimization "Learning and Natural Algorithms"*, *Ph.D. thesis*. Dipartimento di elltronica, Politecnico di Milano
- Dorigo, M., Maniezzo, V., dan Colorni, A. (1999). *"Ant Algorithms for Discrete Optimization"* *Artificial Life*, 5(2):137-172
- Dunn, J. C. (1973) *A Fuzzy Relative of The ISODATA Process and its use in Detecting Compact, Well-Separated Clusters*. Cybern vol.3 hal 32-57
- Gath, I. and Geva A. B. (1989). *Unsupervised Optimal fuzzy clustering*. IEEE Transaction on Pattern Analysis Machine Intelligence 11 hal. 773-781.

- Gonzalez, R. C. Dan Woods, R. E. (2002) *Introduction in Digital Image Processing*. New Jersey, United States of America: Prentice Hall, hal 1-2.
- Gu, T. and Dubuission B. (1990). *Similarity of classes and fuzzy clustering*. Fuzzy Sets and Systems 34, hal. 213-221.
- Hakim, Ardil A., (2006). *Permasalahan Serta Penanggulangan Tumor Otak dan Sumsum Tulang Belakang*. Universitas Sumatra Utara.
- Handayani, L. (2012). *Identifikasi Area Kanker Ovarium pada Citra CT Scan Abdomen menggunakan Metode Expectation Maximization*. Seminar Nasional Teknologi Informasi, Komunikasi dan Industri (SNTIKI): Pekanbaru.
- Haralick, K (1973). *Textural Features for Images Classification*. IEEE Transaction on Systems, Man and Cybernetics, Vol. SMC-3, No. 6
- Izakian, H. dan Abraham, A (2011). *Fuzzy C-Means dan Fuzzy Swarm for Fuzzy Clustering Problem, Expert Systems with Applications*, 38, hal 1835-1838.
- Kadir A. dan Susanto A. (2013). *Teori dan Aplikasi Pengolahan Citra*. Penerbit Andi, Yogyakarta.
- Krishnan, P. H. dan DR. Ramamoorthy, P. (2014). *Fuzzy Clustering based Ant Colony Optimization Algorithm for MR Brain Image Segmentation*. Research Scholar, Department of Electrical Engineering, SNS College of Technology, Coimbatore.
- Lohmann, G. (1998). *Extracting Line Representations of Suical and Gyrat Patterns in MR Images of the Human Brain*. IEEE Transactions on Medical Imaging, Vol 17, No. 6, hal. 1040-1948
- Lhotská, L., Macaš, M., dan Burša, M. (2006). *PSO and ACO in Optimization Problems*. Gersner Laboratory, Czech Technical university in Prague Technicka, 166 27, Prague 6, Czech Republic

- Malik, U. (2009). *Medical Image Segmentation Using Genetic Algorithm*. IEEE TRANSACTION ON INFORMATION TECHNOLOGY IN BIOMEDICINE, vol 13, no 2. hal 166-173
- Munir, R. (2004). *Pengolahan Citra Digital*. Bandung: Informatika
- Mustapha, N (2009). *Expectation Maximization Clustering Algorithm for User Modeling in Web Usage Mining Systems*. European Journal of Scientific Research ISSN 1450-216X Vol.32 No.4
- Nurhayanti (2002). *Deteksi Kelainan Otak Ischemic Cerebral Infaction dari Gambaran Magnetic Resonance Imaging dengan Menggunakan Jaringan Syaraf Tiruan*. Teknik Fisika, Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember.
- Pang, W., Wang, K., Zhou, C., dan Dong, L., (2004). *Fuzzy Discrete Particle Swarm Optimization For Solving Traveling Salesman Problem*. IEEE CS Press, Proceedings of the fourth international conference on computer and information technology hal. 796–800. Price, Sylvia A dan Lorraine, Wilson M., (2006). “*Patofisiologi konsep klinis proses-proses penyakit*”, terjemahan dr. Bram dkk, edisi 6, Vol.2, hala 1171-1174 dan 1183-1186, Buku Kedokteran EGC, Jakarta.
- Pena, J. M., Lozano, J. A. and Larranaga, P. (1999). *An empirical comparison of four initialization methods for the k-means algorithm*. *Pattern Recognition Lett.*, 20:hal. 1027- 1040.
- Piater, J. H. (2002). *Mixture Model and Expectation Maximization*. Lecture at ENSIMAG
- Pradnyana, I. P. G. S. (2015). *Perancangan Sistem Pendeteksi Genangan Air Potensi Perkembangbiakan Nyamuk Melalui Foto Citra Udara Dengan Metode Gray Level Co-Occurence Matrix (GLCM)*, Bandung: Tekom University.
- Prasetyo, E. (2011). *Pengolahan Citra Digital dan Aplikasinya Menggunakan Matlab*. Yogyakarta. ANDI

- Rohani (2013). *Identifikasi Area Tumor pada Citra CT Scan Tumor Otak menggunakan Metode Expectation Maximization Gaussian Mixture Model (EM-GMM)*. Tugas Akhir Teknik Informatika Riau:Universitas Islam Negeri Sultan Syarif Kasim
- Rouben, M. (1982). *Fuzzy Clustering Algorithms and their Clustering Validity*. Europe Journal of Operational Research 10 hal. 385-396
- Selvi, V dan Dr. Umarani, R (2010). *Comparative Analysis of Ant Colony and Particle swarm Optimization Techniques*.
- Shan Shen, William Sandham, Malcolm Granat, Annette Sterr, “MRI Fuzzy Segmentation of Brain Tissue Using Neighborhood Attraction With Neural-Network Optimization”, IEEE transactions on information technology in biomedicine, vol. 9, no. 3, september 2005
- Shih, F. Y. (2010). *Image Processing and Pattern Recognition: Fundamentals and Techniques*. IEEE press-John Wiley & Sons, Inc., Hoboken, New Jersey.
- Soffieti, R. (2003). “Metastasis Brain Tumors”. *7th Congress of The European Federation of Neurological Societies*. Helsinki.
- Sonka, M., Hlavac, V., dan Boyle, R. (2008). *Optimal Thresholding in Image Processing Analysis and Machine Vision International Student Edition*, Hilda Gowanns, Ed. United States of America: Thompsom Corporation, ch 6, hal 180.
- Tan, P., Steinbach, M., dan Kumar, V. (2006) *Introduction to Data Mining*. New York: P bearson education
- Tian, D dan Fan, L (2007). “A Brain MR Images Segmentation Method Based on SOM Neural Network”, ICBBE, hal. 686-689
- Trauwaert, E. (1985). *On The Meaning of Dunn’s Partition Coefficient for Fuzzy Clusters*. International Working Paper, Vrije Universiteit Brussels.

- Xu, Rui dan Donald, W (2009). “*Clustering*”. IEEE Press-John Wiley & Sons, Inc. Hoboken, New Jersey. ISBN: 978-0-470-27680-8.
- Zarpak, B. dan Rahman F. (2008). “Image Segmentation Using Gaussian Mixture Models”, *IUST International Journal of Engineering Science*, Vol. 19, No.1-2, hal. 29-32.
- Zhang, Y., Huang, D., Ji, M., dan Xie, F. (2011), *Image segmentation using PSO and PCM with Mahalanobis distance*. Expert Syst. Appl. 38 hal. 9036–9040.
- Zhang, Y. Y., Brady, M., dan Smith M., (2001). *Segmentation of Brain MR Images through a Hidden Markov Random Field Model and the Expectation Maximization Algorithm*. IEEE Transactions on Medical Imaging, Vol. 20 no. 1

Lampiran

Lampiran 1 syntax preprocessing main.m

```
[filename, pathname]=uigetfile({'*.dcm'});
input=dicomread([pathname, filename]);
K=7;
imshow(input,[]);
[treshold]=Treshold(input);
[head_contour,head_mask]=ContourAndMask(treshold);
[subtract,openingAwal,Detection,closing,otak,hol,open2,preprocessin
g]=BrainArea(head_mask,treshold,input);
Y = mat2gray(input);
imwrite(Y,'inputCont.jpg','Quality',100,'Mode','lossy');
imshow(Y,[]);
```

Lampiran 2 syntax preprocessing thresholding.m

```
function [treshold]=Treshold(input)
treshold=input;
ukuranGambar = size(input)
input=input(:);
t_nol=mean(input);
input=input(:);
G=input;
L=input;
G(G<t_nol)=[];
L(L>t_nol)=[];
sum=mean(G)+mean(L)
t_satu=sum/2;
while(true)
    old_t=t_satu;
    G=input;
    L=input;
    G(G<=old_t)=[];
```

Lampiran 2 syntax preprocessing main.m (lanjutan)

```

    L(L>old_t)=[];
    sum=mean(G)+mean(L);
    t_satu=sum/2;
    if(abs(old_t-t_satu)<0.1)
        break;
    end
end
for i=1:ukuranGambar(1)
    for j=1:ukuranGambar(2)
        if(treshold(i,j)>=t_satu)
            treshold(i,j)=1;
        else
            treshold(i,j)=0;
        end
    end
end
treshold=double(treshold);
imwrite(treshold,'treshold.jpg','Quality',100,'Mode','lossy');
imshow(treshold,[]);

```

Lampiran 3 syntax preprocessing contourandmask.m

```

function [head_contour,head_mask]=ContourAndMask(treshold)

ukuranGambar=size(treshold);
head_contour=zeros(ukuranGambar(1),ukuranGambar(2));
head_mask=zeros(ukuranGambar(1),ukuranGambar(2));
for i=1:ukuranGambar
    [r,c]=find(treshold(i:i,1:ukuranGambar(2)));
    kiri=min(c);
    kanan=max(c);
    head_contour(i,kiri)=1;
    head_contour(i,kanan)=1;

```

Lampiran 3 syntax preprocessing contourandmask.m (lanjutan)

```

imshow(head_contour ,[]);

imwrite(head_contour,'head_contour.jpg','Quality',100,'Mode','lossy');
head_mask(i:i,kiri:kanan)=1;
imshow(head_mask ,[]);
imwrite(head_mask,'head_mask.jpg','Quality',100,'Mode','lossy');
end
imshow(head_mask ,[]);
end

```

Lampiran 4 syntax preprocessing brainarea.m

```

function
[subtract,openingAwal,Detection,closing,otak,hol,open2,preprocessin
g]=BrainArea(head_mask,treshold,input)
subtract=head_mask-treshold;
imwrite(subtract,'subtract.jpg','Quality',100,'Mode','lossy');
figure,imshow(head_mask ,[]);
figure,imshow(treshold ,[]);
figure,imshow(subtract ,[]);
openingAwal=bwareaopen(subtract,200);
imwrite(openingAwal,'openingAwal.jpg','Quality',100,'Mode','lossy');
figure,imshow(openingAwal ,[]);
Detection=edge(uint8(openingAwal),'sobel');
figure,imshow(Detection ,[]);
imwrite(Detection,'Detection.jpg','Quality',100,'Mode','lossy');
se=strel('disk',1);
closing=imclose(Detection,se);
figure,imshow(closing ,[]);
ukurangambar=size(openingAwal);

```

Lampiran 4 syntax preprocessing brainarea.m (lanjutan)

```

gabungan=openingAwal;
figure,imshow(gabungan ,[]);
for i=1:ukurangambar(1);
    for j=1:ukurangambar(2)
        if(closing(i,j)==1)
            gabungan(i,j)=1;
        end
    end
end
hol=imfill(gabungan,'holes');
figure,imshow(hol ,[]);
imwrite(hol,'hol.jpg','Quality',100,'Mode','lossy');
otak=hol-gabungan;
figure,imshow(otak ,[]);
imwrite(otak,'otak.jpg','Quality',100,'Mode','lossy');
open2=bwareaopen(otak,200);
figure,imshow(open2 ,[]);
imwrite(open2,'open2.jpg','Quality',100,'Mode','lossy');
preprocessing=input;
preprocessing(~open2)=0;

figure,imshow(preprocessing,[])
sup = mat2gray(preprocessing);
imshow(sup);
I2 = imcrop(sup,[75 50 360 400]);
imshow(I2)
imwrite(sup,'output.jpg','Quality',100,'Mode','lossy');
%imwrite(I2,'outputcropping.jpg','Quality',100,'Mode','lossy');

```


Lampiran 5 syntax menghitung nilai statistika fitur
citratekstureanalysis.m

```
function varargout = Texture_Analysis(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @Texture_Analysis_OpeningFcn, ...
    'gui_OutputFcn', @Texture_Analysis_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function Texture_Analysis_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;
guidata(hObject, handles);
movegui(hObject,'center');
function varargout = Texture_Analysis_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;
function pushbutton1_Callback(hObject, eventdata, handles)
[filename,pathname] = uigetfile({'*.*'});

if ~isequal(filename,0)
```

Lampiran 5 syntax menghitung nilai statistika fitur
citratekstureanalysis.m (lanjutan)

```

Info = imfinfo(fullfile(pathname,filename));
if Info.BitDepth == 1
    msgbox('Citra masukan harus citra RGB atau Grayscale');
    return
elseif Info.BitDepth == 8
    Img = imread(fullfile(pathname,filename));
    axes(handles.axes1)
    cla('reset')
    imshow(Img)
    title('Grayscale Image')
else
    Img = rgb2gray(imread(fullfile(pathname,filename)));
    axes(handles.axes1)
    cla('reset')
    imshow(Img)
    title('Grayscale Image')
end
else
    return
end

handles.Img = Img;
guidata(hObject,handles);

function pushbutton2_Callback(hObject, eventdata, handles)
Img = handles.Img;
pixel_dist = str2double(get(handles.edit1,'String'));
GLCM = graycomatrix(Img,'Offset',[0 pixel_dist; -pixel_dist
pixel_dist; -pixel_dist 0; -pixel_dist -pixel_dist]);

```

Lampiran 5 syntax menghitung nilai statistika fitur
citratekstureanalysis.m (lanjutan)

```
stats =  
graycoprops(GLCM,{'contrast','correlation','energy','homogeneity'});  
  
Contrast = stats.Contrast;  
Correlation = stats.Correlation;  
Energy = stats.Energy;  
Homogeneity = stats.Homogeneity;  
  
data = get(handles.uitable1,'Data');  
data{1,1} = num2str(Contrast(1));  
data{1,2} = num2str(Contrast(2));  
data{1,3} = num2str(Contrast(3));  
data{1,4} = num2str(Contrast(4));  
data{1,5} = num2str(mean(Contrast));  
  
data{2,1} = num2str(Correlation(1));  
data{2,2} = num2str(Correlation(2));  
data{2,3} = num2str(Correlation(3));  
data{2,4} = num2str(Correlation(4));  
data{2,5} = num2str(mean(Correlation));  
  
data{3,1} = num2str(Energy(1));  
data{3,2} = num2str(Energy(2));  
data{3,3} = num2str(Energy(3));  
data{3,4} = num2str(Energy(4));  
data{3,5} = num2str(mean(Energy));  
  
data{4,1} = num2str(Homogeneity(1));  
data{4,2} = num2str(Homogeneity(2));  
data{4,3} = num2str(Homogeneity(3));
```

Lampiran 5 syntax menghitung nilai statistika fitur
citratekstureanalysis.m (lanjutan)

```
data{4,4} = num2str(Homogeneity(4));
data{4,5} = num2str(mean(Homogeneity));

set(handles.uitable1,'Data',data,'ForegroundColor',[0 0 0])

function edit1_Callback(hObject, eventdata, handles)

function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

Lampiran 6 syntax pemberian noise pada citra noise.m

```
I=imread('ax T2.jpg');
%G1 = imnoise(I,'gaussian',0,0.01);
%G2 = imnoise(I,'gaussian',0,0.05);
G3 = imnoise(I,'gaussian',0,0.002);
%S1 = imnoise(I,'salt & pepper',0.01);
%S2 = imnoise(I,'salt & pepper',0.05);
%S3 = imnoise(I,'salt & pepper',0.1);

%imwrite(G1,'G1.jpg','Quality',100,'Mode','lossy');
%imwrite(G2,'G2.jpg','Quality',100,'Mode','lossy');
imwrite(G3,'G3.jpg','Quality',100,'Mode','lossy');
%imwrite(S1,'S1.jpg','Quality',100,'Mode','lossy');
%imwrite(S2,'S2.jpg','Quality',100,'Mode','lossy');
%imwrite(S3,'S3.jpg','Quality',100,'Mode','lossy');
```

Lampiran 7 syntax similarity.m

```

A=importdata('data.csv');
B=importdata('dataS1.csv');
C=A-B;
sum(C(:) == 0)/262144
D=importdata('data.csv');
E=importdata('dataS2.csv');
F=D-E;
sum(F(:) == 0)/262144
G=importdata('data.csv');
H=importdata('dataS3.csv');
I=G-H;
sum(I(:) == 0)/262144

```

Lampiran 8 syntax menghitung nilai silhouette silhouette.m

```

data=importdata('data2C.csv');
idx=importdata('idx10C.csv');
SI=silhouette(data,idx);
SI_cluster=[mean(SI(idx==1)) mean(SI(idx==2)) mean(SI(idx==3))
mean(SI(idx==4)) mean(SI(idx==5))
mean(SI(idx==6)) mean(SI(idx==7))
mean(SI(idx==8)) mean(SI(idx==9))]
SI_all=mean(SI_cluster)

```

Lampiran 9 syntax klastering EM-GMMemgmmklaster.m

```

function [mask,mu,v,p]=EMSeg(ima,k)
%%
%check image
ima=double(imread(ima));
copy=ima;

```

Lampiran 9 syntax klastering EM-GMMemgmmklaster.m (lanjutan)

```

copy2=ima;
ima=ima(:);      % vectorize ima
mi=min(ima);     % deal with negative
ima=ima-mi+1;    % and zero values
m=max(ima);
s=length(ima);

%%
% histogram citra
h=histogram(ima);
x=find(h);
h=h(x);
x=x(:);h=h(:);

%%
% insiasi parameter
mu=(1:k)*m/(k+1);
v=ones(1,k)*m;
p=ones(1,k)*1/k;

% mulai proses

sml = mean(diff(x))/1000;
while(1)
    % Expectation
    prb = distribution(mu,v,p,x);
    scal = sum(prb,2)+eps;
    loglik=sum(h.*log(scal));

    %Maximization
    for j=1:k
        pp=h.*prb(:,j)./scal;
        p(j) = sum(pp);
        mu(j) = sum(x.*pp)/p(j);
        vr = (x-mu(j));
    end
end

```

Lampiran 9 syntax klastering EM-GMMemgmmklaster.m (lanjutan)

```

        v(j)=sum(vr.*vr.*pp)/p(j)+sml; %update variance
    end
    p = p + 1e-3;
    p = p/sum(p);

    % Exit condition
    prb = distribution(mu,v,p,x);
    scal = sum(prb,2)+eps;
        nloglik=sum(h.*log(scal));
        if((nloglik-loglik)<0.0001)
            break;
        end;
    clf
    plot(x,h);
    hold on
    plot(x,prb,'g--')
    plot(x,sum(prb,2),'r')
    drawnow
end
imwrite(plot,'varr2.jpg','Quality',100,'Mode','lossy');

save nloglik
%%
% menghitung mask
mu=mu+mi-1;
s=size(copy);
mask=zeros(s);

for i=1:s(1),
for j=1:s(2),
    for n=1:k
        c(n)=distribution(mu(n),v(n),p(n),copy(i,j));
    end
    a=find(c==max(c));
    mask(i,j)=a(1);

```

Lampiran 9 syntax klastering EM-GMMemgmmklaster.m (lanjutan)

```

end
end

[row,col] = size(mask);
final_img = zeros(row,col);
for i=1:row
    for j=1:col
        if mask(i,j)==1
            final_img(i,j)=0;
        elseif mask(i,j)==2
            final_img(i,j)=120;
        else
            final_img(i,j)=255;
        end
    end
end
figure,imshow(final_img/255,[]);
imwrite(final_img,'a.jpg','Quality',100,'Mode','lossy');

%%
%subjective analysis
orgimg=copy;
im_bin=mask;
[r,c]=find(im_bin==3);
rc=[r,c];
for j=1:(numel(rc)/2)
    copy(r(j),c(j))=0;
end
figure, subplot(2,1,1);
imshow(copy,[]), title('NROI-Non region of interest');
imwrite(copy,'nroi.jpg','Quality',100,'Mode','lossy');
nroi_image=copy;
logimg=imsubtract(orgimg,copy);
roi_image=logimg;
subplot(2,1,2);

```


Lampiran 9 syntax klastering EM-GMMemgmmklaster.m (lanjutan)

```

imshow(logimg,[]), title('ROI-region of interest');
imwrite(logimg,'roi.jpg','Quality',100,'Mode','lossy');

%%
[h,v]=size(im_bin==3);
tumour_area=(1/h)*(1/v)*(sum(sum(im_bin)));
display('area of tumour is :');
display (tumour_area);
D.tumour_area=tumour_area;
save tumour_area
%%
%%
%pixel calculation
[h,v]=size(im_bin);
number_pixel=(sum(sum(im_bin)));
display('number of pixel in the tumor area :');
display (number_pixel);
D.number_pixel=number_pixel;
save number_pixel
%%
varr1 = edge(logimg,'canny');
varr2 = imfuse(copy2,varr1);
fuse_image=varr2;
figure,imshow(varr2), title('subjective analysis');
imwrite(varr2,'varr2.jpg','Quality',100,'Mode','lossy');
ss=im2uint8(logimg)
[r c]= size(ss);
%%
function y=distribution(m,v,g,x)
%%
x=x(:);
m=m(:);
v=v(:);
g=g(:);
for i=1:size(m,1)

```

Lampiran 9 syntax klastering EM-GMMemgmmklaster.m (lanjutan)

```

    d = x-m(i);
    amp = g(i)/sqrt(2*pi*v(i));
    y(:,i) = amp*exp(-0.5 * (d.*d)/v(i));
end
%%
function[h]=histogram(datos)
datos=datos(:);
ind=find(isnan(datos)==1);
datos(ind)=0;
ind=find(isinf(datos)==1);
datos(ind)=0;
tam=length(datos);
m=ceil(max(datos))+1;
h=zeros(1,m);
for i=1:tam,
    f=floor(datos(i));
    if(f>0 & f<(m-1))
        a2=datos(i)-f;
        a1=1-a2;
        h(f) =h(f) + a1;
        h(f+1)=h(f+1)+ a2;
    end;
end;
h=conv(h,[1,2,3,2,1]);
h=h(3:(length(h)-2));
h=h/sum(h);

```

Lampiran 10 Syntax klastering FCM fcmklaster.m

```

close all;
clear all;
filename='output.jpg';
IM=imread(filename);
[maxX,maxY]=size(IM);

IM=double(IM);

```

Lampiran 10 Syntax klastering FCM fcmklaster.m (lanjutan)

```

IMM=cat(3,IM,IM,IM);
org_img=IM;

%%
cc1=0;
cc2=160;
cc3=255;
ttFcm=0;
while(ttFcm<15)

    ttFcm=ttFcm+1

    c1=repmat(cc1,maxX,maxY);
    c2=repmat(cc2,maxX,maxY);
    c3=repmat(cc3,maxX,maxY);

    c=cat(3,c1,c2,c3);

    ree=repmat(0.000001,maxX,maxY);
    ree1=cat(3,ree,ree,ree);

    distance=IMM-c;
    distance=distance.*distance+ree1;

    daoShu=1./distance;

daoShu2=daoShu(:,1)+daoShu(:,2)+daoShu(:,3);

    distance1=distance(:,1).*daoShu2;
    u1=1./distance1;
    distance2=distance(:,2).*daoShu2;
    u2=1./distance2;
    distance3=distance(:,3).*daoShu2;
    u3=1./distance3;

```

Lampiran 10 Syntax klastering FCM fcmklaster.m (lanjutan)

```

ccc1=sum(sum(u1.*u1.*IM))/sum(sum(u1.*u1));
ccc2=sum(sum(u2.*u2.*IM))/sum(sum(u2.*u2));
ccc3=sum(sum(u3.*u3.*IM))/sum(sum(u3.*u3));

tmpMatrix=[abs(cc1-ccc1)/cc1,abs(cc2-ccc2)/cc2,abs(cc3-
ccc3)/cc3];
pp=cat(3,u1,u2,u3);
for i=1:maxX
    for j=1:maxY
        if max(pp(i,j,:))==u1(i,j)
            IX2(i,j)=1;
        elseif max(pp(i,j,:))==u2(i,j)
            IX2(i,j)=2;
        else
            IX2(i,j)=3;
        end
    end
end

if max(tmpMatrix)<0.0001
    break;
else
    cc1=ccc1;
    cc2=ccc2;
    cc3=ccc3;
end

for i=1:maxX
    for j=1:maxY
        if IX2(i,j)==3
            IMMM(i,j)=250;
        elseif IX2(i,j)==2;
            IMMM(i,j)=160;
        else
            IMMM(i,j)=0;
        end
    end
end

```

Lampiran 10 Syntax klastering FCM fcmklaster.m (lanjutan)

```

        end
    end

end

    for i=1:maxX
        for j=1:maxY
            if IX2(i,j)==3
                IMMM(i,j)=250;
            elseif IX2(i,j)==2;
                IMMM(i,j)=160;
            else
                IMMM(i,j)=0;
            end
        end
    end

    IMMM=uint8(IMMM);
    figure(1),imshow(IMMM);
    imwrite(IMMM,'IMMM3cluster.jpg','Quality',100,'Mode','lossy');

%%
IMMM=uint8(IMMM);
%figure(1)
%imshow(IMMM,[]);
%title('fcm output');
disp('The final cluster centers are');
ccc1
ccc2
ccc3

%%
% daerah ROI dan NROI
im_bin=IX2;
[r,c]=find(im_bin==3);
rc=[r,c];

```

Lampiran 10 Syntax klastering FCM fcmklaster.m (lanjutan)

```



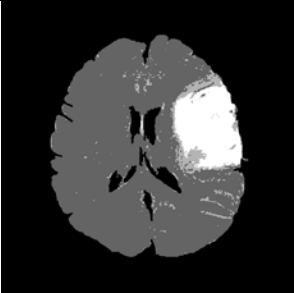
simg=IM;

im=im2uint8(IM);
max_val = max(max(simg));
z = max_val/255;
simg = simg./z;
simg=uint8(simg);
orgimg=simg;
for j=1:(numel(rc)/2)
    simg(r(j),c(j))=0;
end
figure, subplot(2,1,1);
imshow(simg,[]), title('NROI-Non region of interest');
nroi_image=simg;
logimg=imsubtract(orgimg,simg);
roi_image=logimg;
subplot(2,1,2);
imshow(logimg,[]), title('ROI-region of interest');
%imwrite(logimg,'ROI3cluster.jpg','Quality',100,'Mode','lossy');
%%
%area calculation
[h,v]=size(im_bin);
tumour_area=(1/h)*(1/v)*(sum(sum(im_bin)));
%%
%pixel calculation

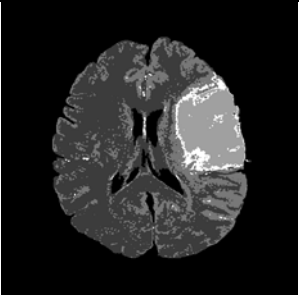
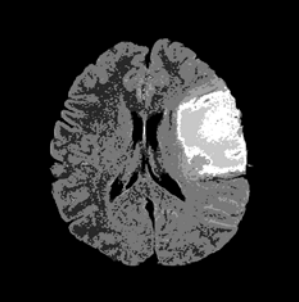
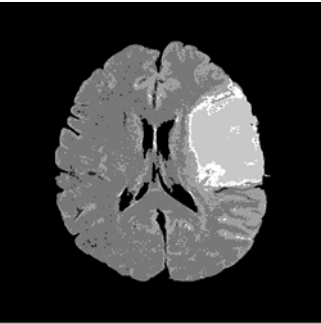
%%
%subjective analysis
var1 = edge(im_bin==3,'canny');
var2 = imfuse(org_img,var1);
fuse_image=var2;
figure, imshow(var2), title('subjective analysis');
imwrite(var2,'var23cluster.jpg','Quality',100,'Mode','lossy');

```

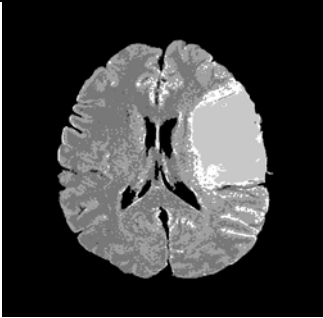
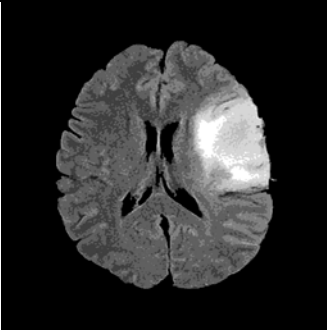
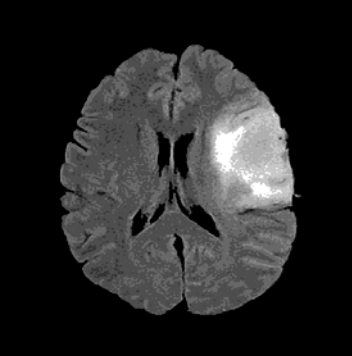
Lampiran 11 Hasil Segmentasi Citra *ax T2 flair* metode EM-GMM

Jumlah Kluster	Hasil Segmentasi	
2		
3		
4		

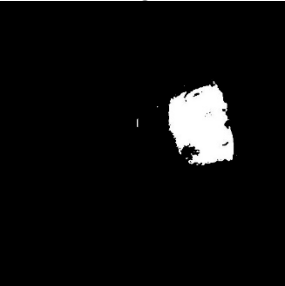

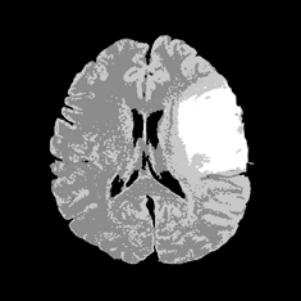
Lampiran 11 Hasil Segmentasi Citra *ax T2 flair* metode EM-GMM
(lanjutan)

5	
6	
7	


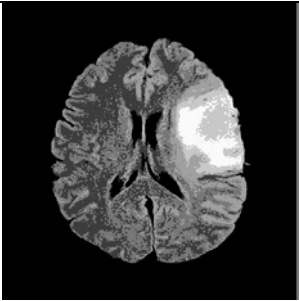
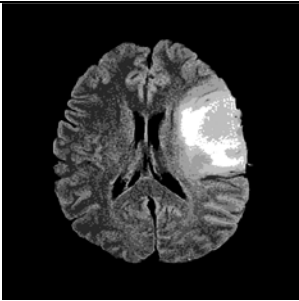
Lampiran 11 Hasil Segmentasi Citra *ax T2 flair* metode EM-GMM
(lanjutan)

8	
9	
10	

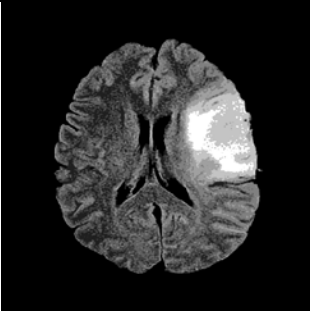
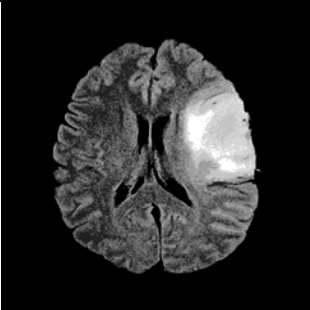
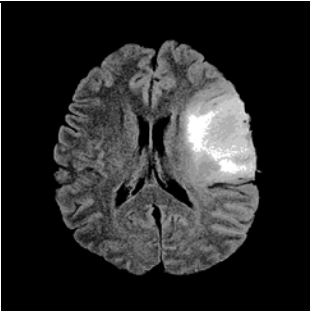
Lampiran 12 Hasil Segmentasi Citra *ax T2 flair* metode FCM

Jumlah Klaster	Hasil Segmentasi	
2		
3		
4		


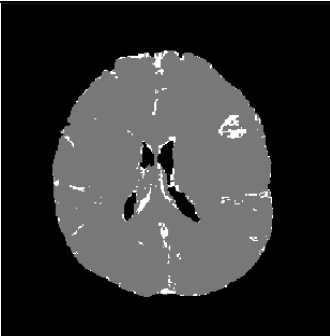
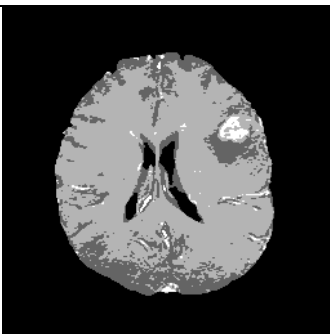
Lampiran 12 Hasil Segmentasi Citra *ax T2 flair* metode FCM
(lanjutan)

5	
6	
7	

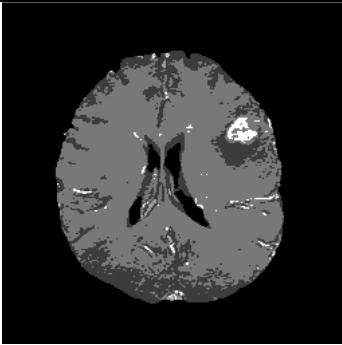
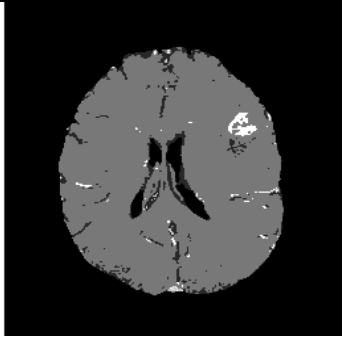
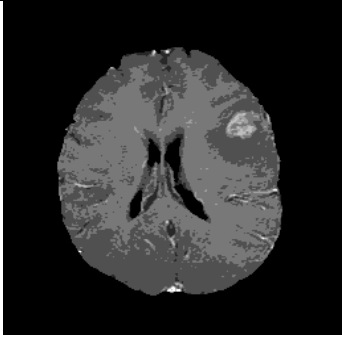
Lampiran 12 Hasil Segmentasi Citra *ax T2 flair* metode FCM
(lanjutan)

8	
9	
10	

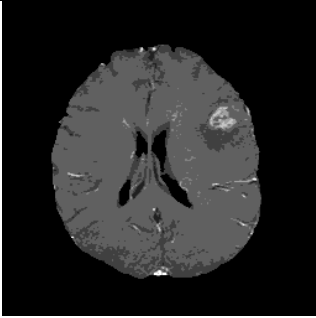

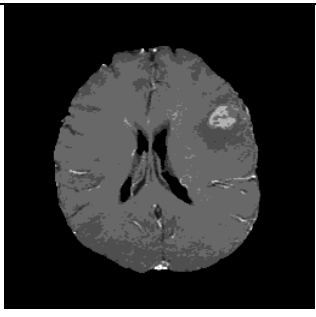
Lampiran 13 Hasil Segmentasi Citra *ax* T1 *memp+c* metode EM-GMM

Jumlah Kluster	Hasil Segmentasi	
2		
3		
4		




Lampiran 13 Hasil Segmentasi Citra *ax* T1 *memp+c* metode EM-GMM (lanjutan)

5	
6	
7	

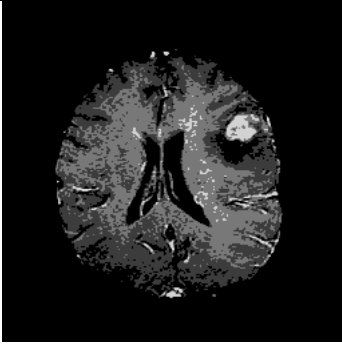
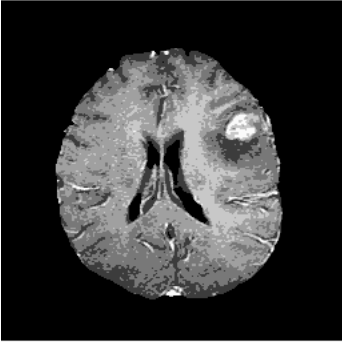
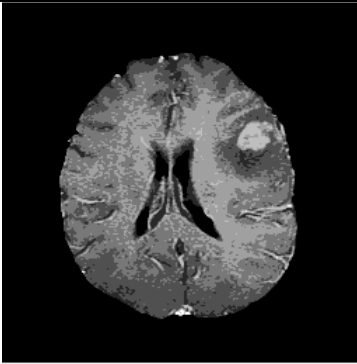
Lampiran 13 Hasil Segmentasi Citra *ax* T1 *memp+c* metode EM-GMM (lanjutan)

8	
9	
10	

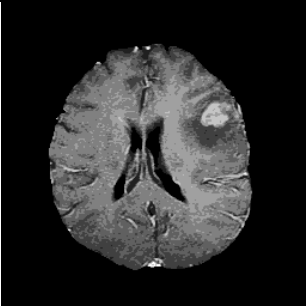

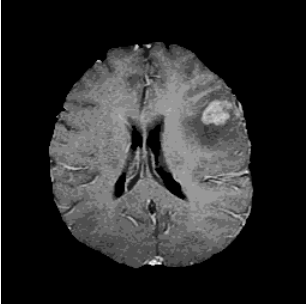
Lampiran 14 Hasil Segmentasi Citra *ax* T1 *memp+c* metode FCM

Jumlah Klaster	Hasil Segmentasi	
2		
3		
4		

Lampiran 14 Hasil Segmentasi Citra *ax T1 memp+c* metode FCM
(lanjutan)

5	
6	
7	

Lampiran 14 Hasil Segmentasi Citra *ax T1 memp+c* metode FCM
(lanjutan)

8	
9	
10	

Lampiran 15 Nilai *silhouette index*

Jumlah klaster	Data	
	Nilai silhouette ax T2 <i>flair</i>	Nilai silhouette ax T1 <i>memp+c</i>
2	0.8307	0.8972
3	0.9565	0.9295
4	0.9275	0.9056
5	0.916	0.8988
6	0.9075	0.8959
7	0.9007	0.8953
8	0.8956	0.8948
9	0.8937	0.8927
10	0.8902	0.8904

Lampiran 16 Nilai *partition coefficient index*

Jumlah klaster	Data	
	Nilai <i>partition coefficient index</i> ax T2 <i>flair</i>	<i>partition coefficient index</i> ax T1 <i>memp+c</i>
2	0.9047	0.9424
3	0.9118	0.4726
4	0.828	0.3476
5	0.7043	0.4952
6	0.6666	0.4313
7	0.7526	0.6881
8	0.7447	0.6439
9	0.7318	0.6377
10	0.6916	0.573

Lampiran 17 Nilai *similarity* data *ax T2 flair* metode EMGMM

EMGMM						
Jumlah Klaster	salt and pepper			gaussian		
	1 persen	5 persen	10 persen	1 persen	5 persen	10 persen
2	0.9951	0.9749	0.9503	0.7198	0.6856	0.6619
3	0.9935	0.9669	0.9345	0.7007	0.4643	0.4646
4	0.9933	0.9666	0.9338	0.4009	0.4185	0.4197
5	0.9928	0.9647	0.9309	0.463	0.4243	0.4185
6	0.9928	0.9641	0.9302	0.5296	0.45	0.4291
7	0.9929	0.9648	0.9313	0.5368	0.452	0.4237
8	0.9924	0.9642	0.9303	0.5123	0.4275	0.4055
9	0.9305	0.9647	0.9924	0.4797	0.4116	0.3904
10	0.9924	0.9642	0.9292	0.4886	0.3961	0.3775
rataan	0.9861	0.9661	0.9403	0.5368	0.4588	0.4434

Lampiran 18 Nilai *similarity* data *ax T2 memp+c* metode FCM

FCM						
Jumlah Klaster	salt and pepper			gaussian		
	1 persen	5 persen	10 persen	1 persen	5 persen	10 persen
2	0.9951	0.9748	0.9504	0.9855	0.9363	0.8969
3	0.9922	0.9625	0.9285	0.9356	0.7163	0.6455
4	0.9844	0.8607	0.789	0.8242	0.6167	0.5613
5	0.993	0.9657	0.9322	0.7487	0.5501	0.4907
6	0.9685	0.933	0.9012	0.6285	0.5228	0.4783
7	0.9738	0.9342	0.9021	0.5934	0.4839	0.4494
8	0.9691	0.9277	0.8957	0.5662	0.4621	0.4276
9	0.9592	0.9313	0.8991	0.558	0.4543	0.4205
10	0.9746	0.9453	0.9125	0.5278	0.434	0.4049
rataan	0.9788	0.9372	0.9011	0.7075	0.5751	0.5305

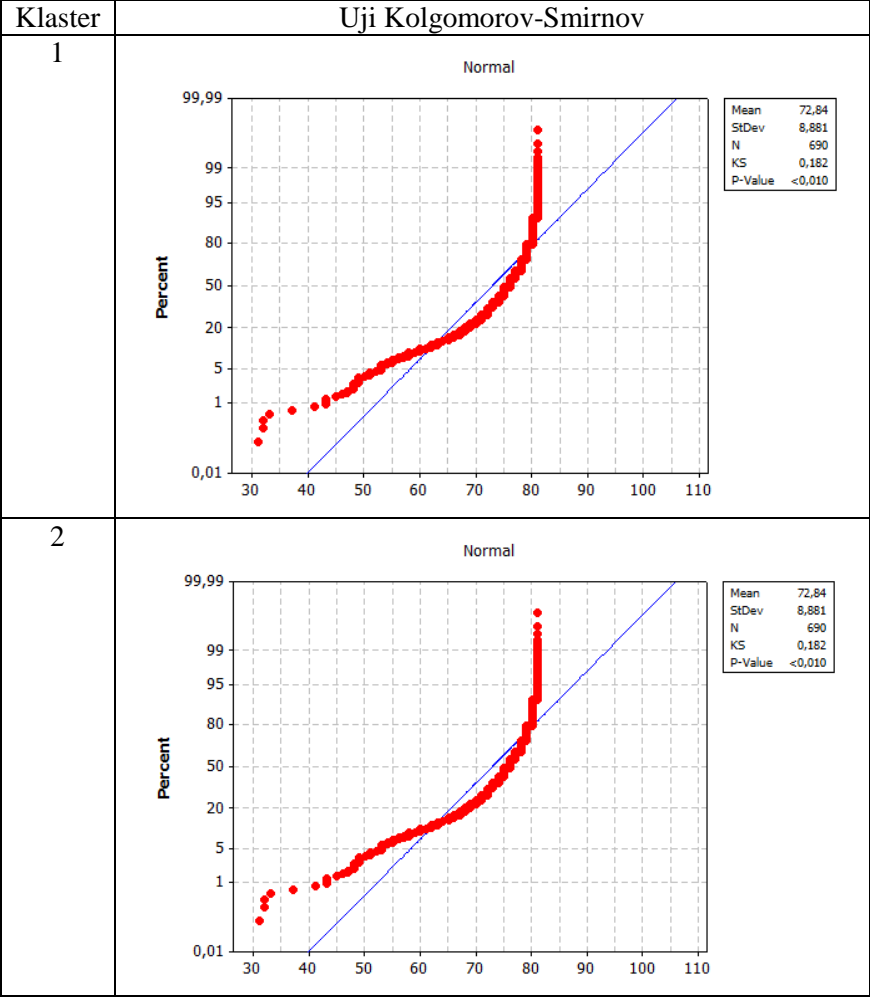
Lampiran 19 Nilai *similarity* data *ax* T1 metode EMGMM

EMGMM						
	salt and pepper			gaussian		
	1 persen	5 persen	10 persen	1 persen	5 persen	10 persen
2	0.9944	0.9745	0.9525	0.7247	0.6892	0.6637
3	0.9926	0.9635	0.9277	0.3558	0.4066	0.4253
4	0.9926	0.9458	0.9158	0.3408	0.3676	0.3853
5	0.992	0.956	0.9211	0.6109	0.4593	0.4434
6	0.9924	0.9648	0.9924	0.557	0.4631	0.4319
7	0.9917	0.9917	0.9632	0.5163	0.4325	0.4034
8	0.992	0.9616	0.9291	0.5107	0.4154	0.3881
9	0.9914	0.9477	0.9166	0.4866	0.3991	0.3812
10	0.9916	0.9598	0.9279	0.4695	0.3859	0.3687
rataan	0.9923	0.9628	0.9384	0.5080	0.4465	0.4323

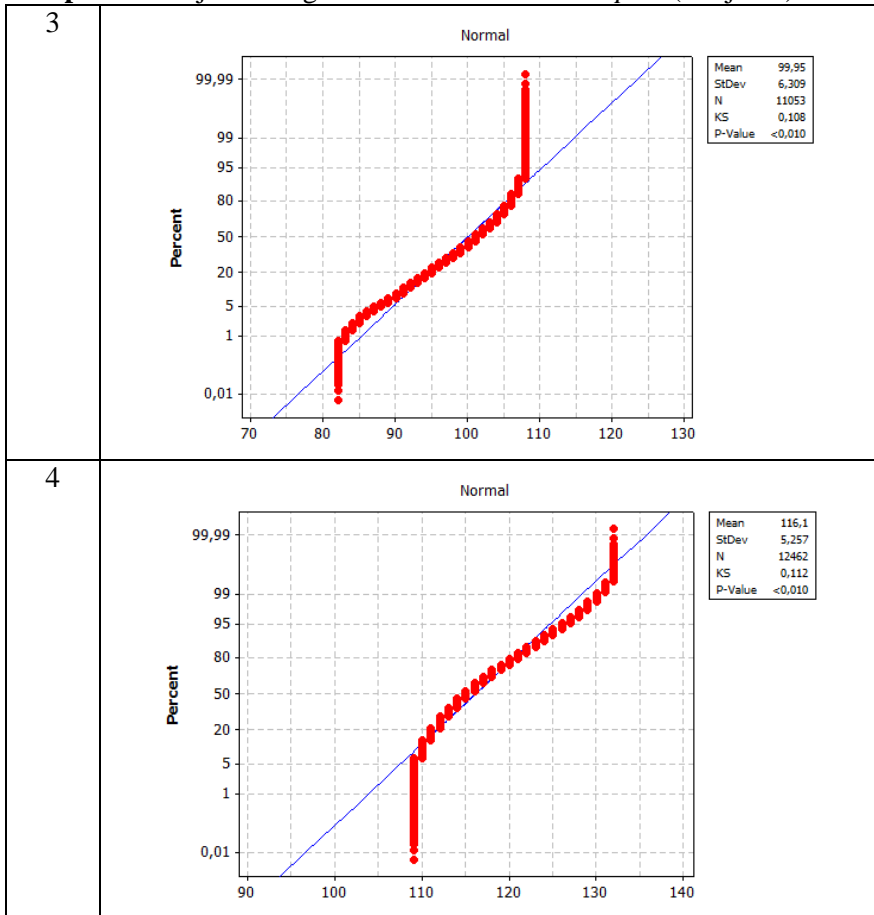
Lampiran 20 Nilai *similarity* data *ax* T1 *memp+c terhadap salt and pepper noise*

FCM						
	salt and pepper			gaussian		
	1 persen	5 persen	10 persen	1 persen	5 persen	10 persen
2	0.9948	0.9948	0.9521	0.9912	0.9912	0.9912
3	0.7889	0.7749	0.7594	0.8426	0.8426	0.6274
4	0.8658	0.8452	0.8215	0.7306	0.5837	0.5362
5	0.9925	0.9653	0.9653	0.7644	0.5456	0.4836
6	0.9229	0.9146	0.8664	0.5855	0.5065	0.4672
7	0.9898	0.9619	0.9287	0.5761	0.476	0.4388
8	0.9902	0.9607	0.9193	0.5511	0.4523	0.4214
9	0.9914	0.9477	0.9166	0.4866	0.3991	0.3812
10	0.9891	0.9611	0.9223	0.5148	0.4275	0.4008
	0.94727	0.9251	0.8946	0.6714	0.5805	0.5275

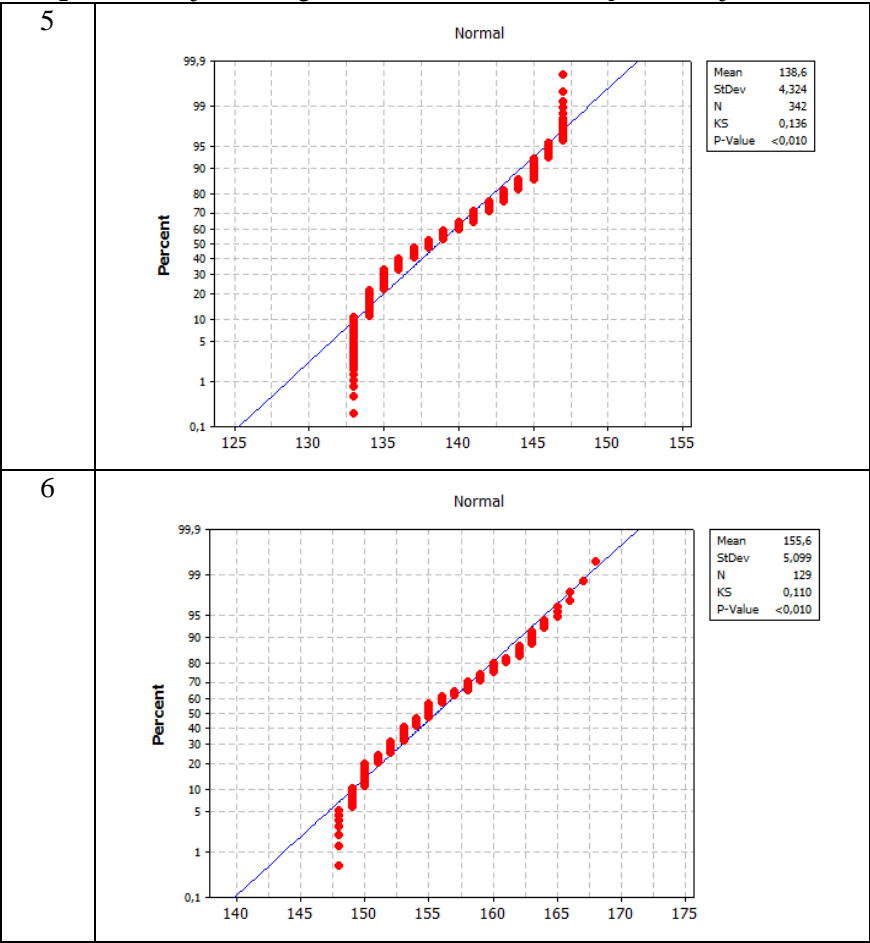
Lampiran 21 Uji Kolmogorov Smirnov *ax Tl memp+c*

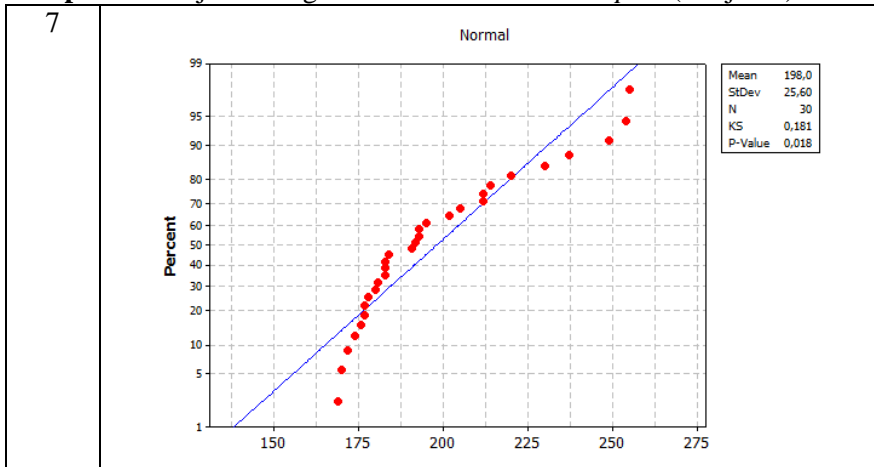
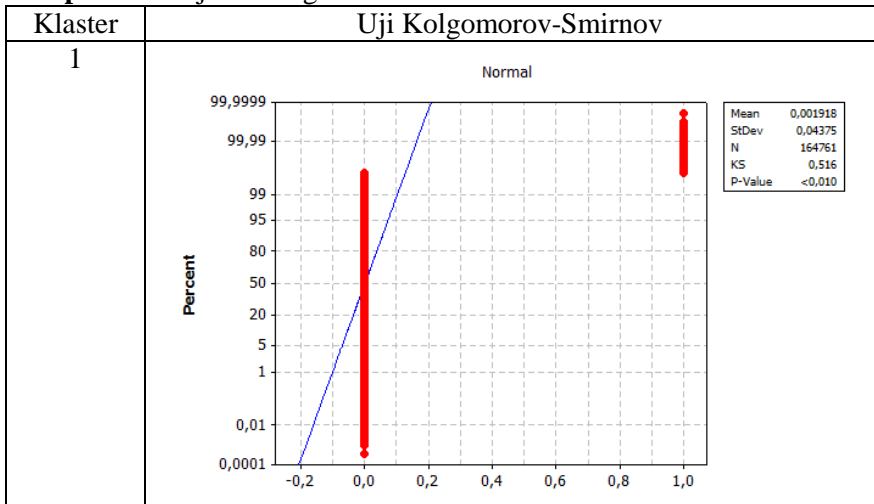


Lampiran 21 Uji Kolmogorov Smirnov *ax Tl memp+c* (Lanjutan)

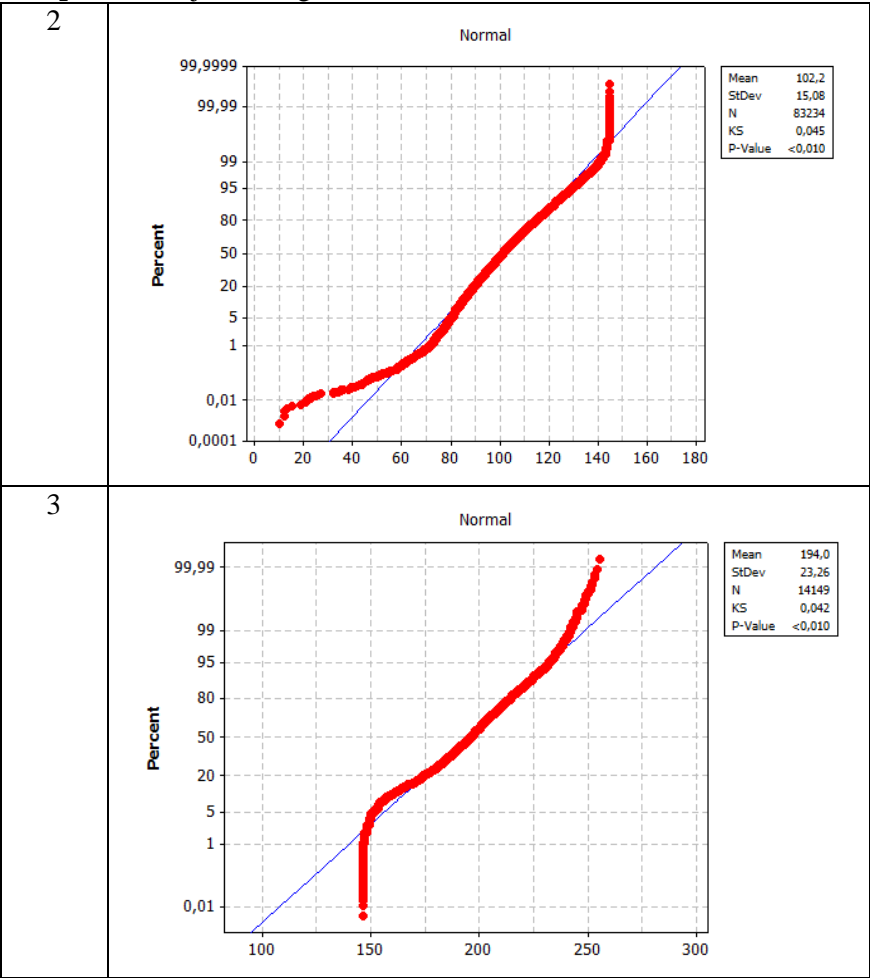


Lampiran 21 Uji Kolmogorov Smirnov ax T1 memp+c (Lanjutan)



Lampiran 21 Uji Kolmogorov Smirnov $ax T1 memp+c$ (Lanjutan)

Lampiran 22 Uji Kolmogorov Smirnov $ax T2 Flair$


Lampiran 22 Uji Kolmogorov Smirnov *ax T2 Flair*



BIODATA PENULIS



Penulis memiliki nama lengkap Win Heber Goklas Sianipar atau biasa dipanggil dengan nama Win. Penulis lahir di Jakarta pada tanggal 5 April 1995, namun sudah tinggal dan menempuh pendidikan wajib di Tangerang Selatan sejak berusia 3 tahun. Penulis merupakan putra dari pasangan Bapak Sianipar dan Ibu Dr. Gultom . Penulis menempuh pendidikan SD di SDS STRADA

BHAKTI UTAMA sedangkan pendidikan SMP di SMP Negeri 161 Jakarta Selatan. Kemudian penulis menempuh pendidikan SMA di SMA Negeri 90 Jakarta. Hingga akhirnya pada tahun 2013 penulis melanjutkan pendidikan di jenjang perguruan tinggi di jurusan Statistika ITS melalui jalur MANDIRI. Selama 4 tahun berkuliah di jurusan Statistika ITS, penulis juga aktif di beberapa organisasi, antara lain adalah PSt-HIMASTA-ITS 2014-2015, Mahasiswa Bonapasogit ITS 2014-2016, Data Science Indonesia Regional Jawa Timur dan Python Surabata (surabaya.py). Selama aktif di organisasi tersebut penulis pernah menjabat sebagai Manager PSt Himasta ITS pada masa kepengurusan 2015. Segala kritik dan saran serta diskusi lebih lanjut mengenai Tugas Akhir ini dapat dikirimkan melalui surat elektronik (*e-mail*) ke sianiparheber@gmail.com atau nomor telepon 087855632560.